

计 算 机 科 学 丛 书

数字图像处理 疑难解析

(英) Maria Petrou (希) Panagiota Bosdogianni 著 赖剑煌 冯国灿 等译

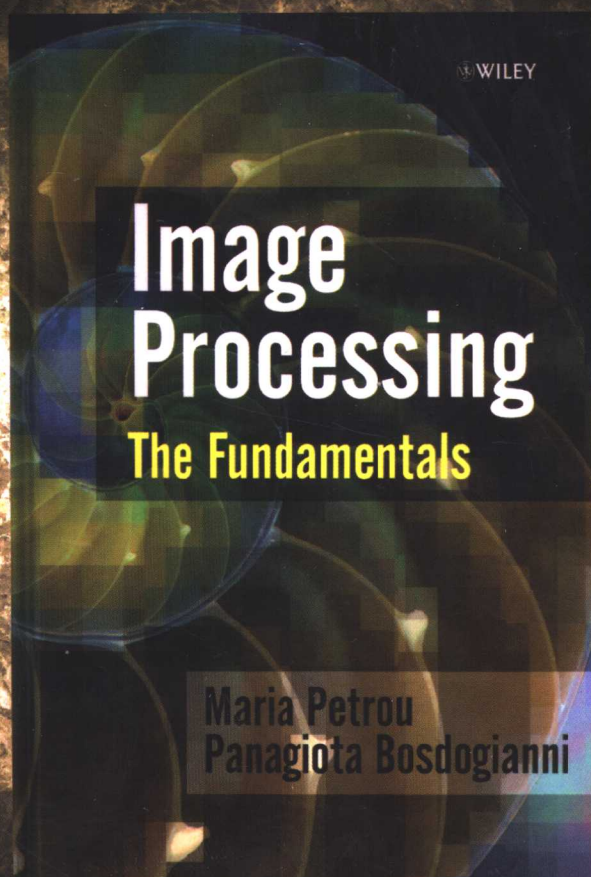


Image Processing
The Fundamentals



机械工业出版社
China Machine Press

本书作者结合在英国Surrey大学从事十多年“信号处理与机器智能”研究生课程教学的经验，对图像处理的基本原理做了深入浅出的论述。全书用例详实，实用性强。

本书特点：

- 通过用尺度小的图像来解释概念，便于进行运算
- 深入阐述每种方法的细节，包括问题来源以及在实践中可能会遇到的问题
- 给出了深、浅两个层次的详细数学解释

ISBN 7-111-15544-0



华章图书

华章网站 <http://www.hzbook.com>

网上购书：www.china-pub.com

北京市西城区百万庄南街1号 100037

读者服务热线：(010)68995259, 68995264

读者服务信箱：hzedu@hzbook.com

ISBN 7-111-15544-0/TP · 4053

定价：33.00 元

计 算 机 科 学 丛 书

数字图像处理 疑难解析

(英) Maria Petrou (希) Panagiota Bosdogianni 著 赖剑煌 冯国灿 等译

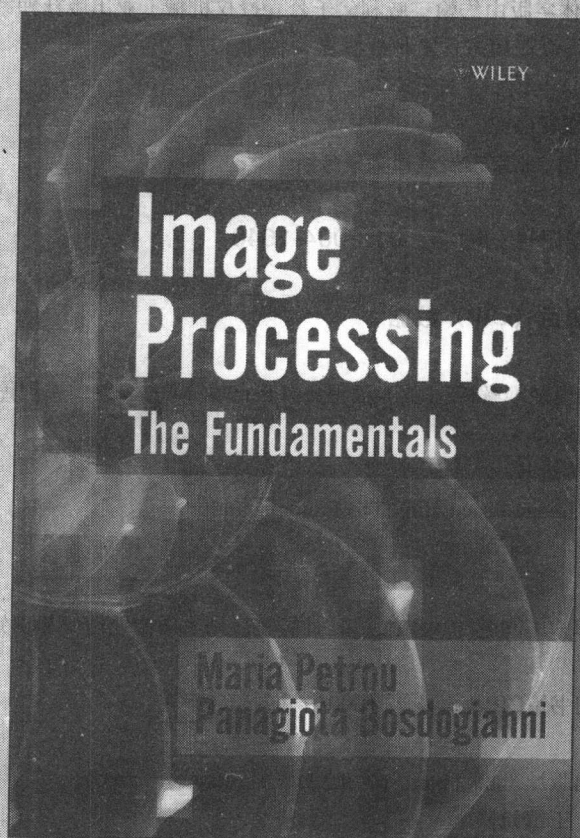


Image Processing
The Fundamentals



机械工业出版社
China Machine Press

本书通过问答形式介绍数字图像处理的基础理论, 涉及图像处理技术的各个层面。
本书作者有丰富教学经验, 全书用例详实, 内容深入浅出, 实用性强。

本书可作为大学计算机及相关专业本科生、研究生的教学参考书, 也可供广大从事
数字图像处理与应用研究的专业人员参考。

Maria Petrou, et al: Image Processing, The Fundamentals (ISBN: 0-471-99883-4).

Authorized translation from the English language edition published by John Wiley &
Sons, Inc.

Copyright © 1999 by John Wiley & Sons, Inc.

All rights reserved.

本书中文简体字版由约翰·威利父子公司授权机械工业出版社独家出版。未经出版
者书面许可, 不得以任何方式复制或抄袭本书内容。

版权所有, 侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2004-1648

图书在版编目(CIP)数据

数字图像处理疑难解析/(英)彼得鲁(Petrou, M.)等著; 赖剑煌等译. -北京: 机
械工业出版社, 2005.4

(计算机科学丛书)

书名原文: Image Processing, The Fundamentals

ISBN 7-111-15544-0

I. 数… II. ①彼… ②赖… III. 计算机应用-图像处理-基本知识 IV. TP391.41

中国版本图书馆CIP数据核字(2004)第130693号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 王镇元

北京昌平奔腾印刷厂印刷·新华书店北京发行所发行

2005年4月第1版第1次印刷

787mm×1092mm 1/16·17.5印张

印数: 0 001 - 4 000册

定价: 33.00元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换
本社购书热线: (010) 68326294

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅肇划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及度藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业

的教学度身订造的。其中许多教材均已为M. I. T., Stanford, U.C. Berkeley, C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程,而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下,读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑,这些因素使我们的图书有了质量的保证,但我们的目标是尽善尽美,而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正,我们的联系方法如下:

电子邮件: hzedu@hzbook.com

联系电话: (010) 68995264

联系地址: 北京市西城区百万庄南街1号

邮政编码: 100037

专家指导委员会

(按姓氏笔画顺序)

尤晋元	王 珊	冯博琴	史忠植	史美林
石教英	吕 建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周立柱	周克定	周傲英	孟小峰	岳丽华
范 明	郑国梁	施伯乐	钟玉琢	唐世渭
袁崇义	高传善	梅 宏	程 旭	程时端
谢希仁	裘宗燕	戴 葵		

秘 书 组

武卫东

温莉芳

刘 江

杨海玲

译者序

数字图像处理是一门新兴学科，它起源于20世纪20年代，但是直到80至90年代，随着计算机应用的发展，数字图像处理才逐渐形成一套独立的学科体系。近20年来，由于遥感、医学、军事、工业和通信等领域应用的需要，数字图像处理的科学理论与技术手段得到迅猛发展，已成为了一个跨越计算机科学、数学、遥感、气象、冶金、地球物理学、生物医学等多学科的前沿科技领域。从CT的发明、数码相机的普及和数字电视业务的开展，到遥感图像处理、生物特征鉴别和智能交通的应用，数字图像处理的应用随处可见，它已极大地促进了人类的科学研究的发展、社会生产率的提高和生活方式的改善。因此，作为一个有广阔应用前景的新兴学科，无论是在理论研究方面，还是在应用生产方面，数字图像处理目前都存在许多问题有待我们去探索。

数字图像处理的核心是数学理论，图像的采集、变换、表达、增强、滤波、复原和分割，无一不与数学密切相关。对于初学者来说，学习数字图像处理这门学科往往会有许多疑问，也会因为表面深奥的数学公式而裹足不前。本书刚好可以解决这些问题。

这本书是作者在数字图像处理领域11年教学工作的成果，它有如下特色：

- 写作形式独特。它以提问和回答的形式循序渐进、深入浅出地阐述了图像处理的基本原理。其中许多问题都来源于教学中学生的提问，很有代表性。这些问题往往是读者都会遇到的。
- 对数学的处理恰到好处。它给出了两个层次的详细数学解释：一个层次是为了便于看懂而做的叙述，尽可能少地涉及数学；另一个是高层次的，严格的数学证明。
- 对图像处理问题的来龙去脉解释清楚。使读者能够理解每种方法的“基本的细节”，包括有关问题来源及在实践中可能会遇到的问题。
- 通过用尺度小的图像来解释概念，使读者不必借助计算机就可以做运算。
- 分析了大量的实例，指导读者克服难点，获得答案。

本书由中山大学数学与计算科学学院从事数字图像处理的有关师生共同翻译。赖剑煌教授和冯国灿教授负责翻译并审校了全书，其中赖剑煌教授负责本书的第1、5、6、7章，冯国灿教授负责本书的第2、3、4章。参加翻译的还有冯莉、李星、石林英、陈家大、陈羽、郑伟诗、谭丹娜、苏秋斌、何婧、江永宏等。

由于时间仓促，译者水平有限，书中难免存在错误和疏漏，欢迎读者批评指正。

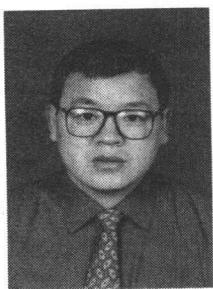
赖剑煌、冯国灿

2004年8月26日

译者简介



赖剑煌，1986年毕业于中山大学，1989年获硕士学位，1999年获博士学位。曾到美国HARRIS数字系统公司、联合国大学软件技术研究所（澳门）等作访问学者。现任中山大学数学系教授、博士生导师、应用数学教研室主任、中国图像图形学会常务理事、广东省图像图形学会常务理事和秘书长。主要研究方向是图像处理和模式识别，尤其是人脸检测与人脸识别技术。迄今，已在国际学术期刊和会议上发表论文50余篇。曾获商业部科技进步二等奖及其他科技奖励多项。



冯国灿，1988年获得中山大学硕士学位，1999年获香港浸会大学计算机科学博士学位。2000年10月~2002年11月获邀请到英国格莱莫根大学（University of Glamorgan）数字图像实验室和布拉德福大学（University of Bradford）数字媒体实验室做博士后研究员，主要从事压缩域图像特征提取和基于内容的图像检索技术的研究。现任中山大学教授、数学系副主任、博士生导师。近年来，主要从事模式识别和计算机视觉的研究，尤其是人脸识别技术和压缩域图像及视频特征提取。迄今，已在国际学术期刊和会议上发表论文50余篇。

前言

本书是作者给Surrey大学信号处理和机器智能专业的硕士研究生开设本门课程11年来的教学成果的总结。

图像处理作为机器智能的一门学科，最近几年得到了蓬勃发展，因此，引起了来自心理学、生理学、工程学和数学等不同领域研究人员的关注。机器智能的各种问题可以用不同的工具来解决。但是，当我们要用计算机来完成一项任务时，就必须要用机器所能理解的语言，而且这种语言是一种从数学领域发展出来算术语言，因此解决机器智能问题的方法主要是数学方法。图像处理是所有与机器智能相关的视觉和图像研究的基础。那种撇开图像处理去实现计算机视觉的想法就像希望从屋顶开始建造房子一样是不现实的，而不借助数学来做图像处理就像挥动胳膊去飞翔一样也是不可能的。

由于打算在机器智能领域工作的人员层次不同，所以本书分为两个层次。第一个层次用有限的数学知识就很容易地理解，这个层次适合于那些进入这个领域的新手和研究生。第二个层次就比较复杂，需要用到各种复杂的数学方法和证明，适合于那些不屈不挠地刨根问底，想要知道“为什么”和“怎么样”的有好奇心的学生。对较低层次的读者，阅读本书时可以不参考为高层次读者所写的内容。所有涉及到高层次的内容都放在方框里，可以忽略。本书包含了大量例子。与方框中内容相联系的例子都用大写字母B标记出来了，如果愿意，它们可以和那些高级内容一样忽略。读者需要具备的基本的数学背景知识是如何进行矩阵的加减运算，矩阵的特征值分析的知识也是很重要的。但是，本书会有几个完全可行的例子，使得即使不熟悉这些知识的人，也能通过本书很容易地学会这种基本要点。本书介绍的随机方法也一样：读者可以从随机变量的基本概念开始学习，然后达到理解和运用遍历概念的水平。

借此机会我要感谢无数的硕士生这么多年帮助我完善这本书，他们尖锐的提问，或者有时候看起来天真的问题，都对我有很大帮助。其实，人们在学习的时候是没有天真的问题的，天真的反而是那些不问问题的人！学生的提问有助于我清楚地规划好学习进度，并给了我内容用提问和回答的形式表示出来的想法。

写这本书对于Panagiota和我来说都是一个学习的过程。通过示例图像我们找到很大的乐趣，并且发现了这些方法的秘密。最困扰我们的一件事就是连续和离散方法之间的差别。一个解析推导得出的适用于连续域的公式通常和计算机编程来完成任务的公式没什么联系。这一点在第6章“图像复原”中会看到清晰的例证。这也是为什么在展示所有我们提出的方法时都使用了小的离散的图像的原因，这使得我们可以人工地操作这些图像，而又能知道当一幅真实大小的图像输入计算机时计算机将如何处理。谈到真实大小的图像，我们要感谢Constantinos Boukouvalas帮助编写了一些方法的程序。

最后，我要感谢Surrey大学的视觉、语音和信号处理中心的同事们，尤其感谢Josef Kittler博士提供给我的所有机会和支持，还有我们的主管Graeme Wilford的热心帮助。

Maria Petrou

目 录

出版者的话	
专家指导委员会	
译者序	
译者简介	
前言	
第1章 导论	1
1.1 为什么要做图像处理	1
1.2 什么是一幅图像	1
1.3 什么是一幅图像在一个像素点的亮度	1
1.4 为什么图像经常用 512×512 、 256×256 、 128×128 等形式表述	2
1.5 存储一幅图像需要多少比特	2
1.6 什么是图像的分辨率	2
1.7 如何进行图像处理	3
1.8 什么是一个线性算子	4
1.9 算子是如何定义的	5
1.10 一个算子是如何对一幅图像进行变换的	5
1.11 什么是点扩散函数	5
1.12 如何表达一个线性算子作用于了一幅图像的实际效果	7
1.13 矩阵H结构的可分离性假设的含义是什么	11
1.14 可分离变换如何用矩阵形式来表示	11
1.15 可分离性假设的含义是什么	12
1.16 本章要点	14
1.17 图像处理的目的是什么	14
1.18 本书要点	15
第2章 图像变换	17
2.1 本章概述	17
2.2 如何定义基图像	17
2.3 什么是两个向量的外积	17
2.4 如何用向量外积进行图像展开	17
2.5 什么是酉变换	19
2.6 什么是酉矩阵	19
2.7 什么是酉变换的逆	19
2.8 如何构造酉矩阵	19
2.9 如何选择矩阵 U 和 V 使得 g 能用比 f 少的比特来表示	19
2.10 如何对角化矩阵	20
2.11 如何计算图像对角化时所需要的矩阵 U 、 V 及 $\Lambda^{\frac{1}{2}}$	24
2.12 什么是图像的奇异值分解	27
2.13 如何用SVD逼近一幅图像	27
2.14 用SVD逼近图像的误差是多少	28
2.15 如何最小化重构误差	29
2.16 用SVD进行图像展开时的基图像是怎样的	30
2.17 任意被展开的图像是否存在基图像集	35
2.18 什么是函数的完全标准正交集	35
2.19 是否存在标准正交离散函数完全集	36
2.20 如何定义哈尔函数	36
2.21 如何定义沃尔什函数	37
2.22 如何从哈尔函数和沃尔什函数创建一个图像变换矩阵	37
2.23 哈尔变换的基图像是怎样的	40
2.24 如何仅用+1或-1定义一个正交矩阵	45
2.25 哈达玛/沃尔什变换的基图像是怎样的	45
2.26 沃尔什和哈尔变换的优缺点是什么	49
2.27 什么是哈尔小波	49
2.28 傅里叶变换的离散形式是怎样的	49
2.29 离散傅里叶变换如何用矩阵形式表示	51
2.30 矩阵 U 可否用于DFT的酉矩阵	52
2.31 用DFT进行图像展开时的基图像是怎样的	53
2.32 为什么离散傅里叶变换比其他变换使用更普遍	56

2.33 什么是卷积定理	57	3.22 遍历性的含义是什么	79
2.34 如何显示图像的离散傅里叶变换	62	3.23 如何利用遍历性来减少表示一幅 图像所需要的比特数	79
2.35 图像旋转时的离散傅里叶变换是 怎样的	63	3.24 具有不相关随机变量的随机场的 自相关函数的形式是怎样的	79
2.36 图像平移时的离散傅里叶变换是 怎样的	64	3.25 如何变换图像使得它的自相关矩阵 是对角的	80
2.37 函数平均值和它的DFT之间的关系 是怎样的	65	3.26 遍历性的假设是现实的吗	80
2.38 图像伸缩时的离散傅里叶变换是 怎样的	66	3.27 如何用K-L变换来逼近一幅图像	85
2.39 什么是离散余弦变换	68	3.28 当我们截断K-L扩展来逼近一幅 图像时的误差是怎样的	85
2.40 本章要点	68	3.29 K-L变换的基图像是怎样的	85
第3章 图像的统计描述	69	3.30 本章要点	96
3.1 本章概述	69	第4章 图像增强	97
3.2 为什么需要对图像进行统计描述	69	4.1 什么是图像增强	97
3.3 是否存在一个用不相关数据来表示 的图像变换可以按最小均方误差来 近似图像	69	4.2 如何增强图像	97
3.4 什么是随机场	69	4.3 利用图像的灰度级统计进行图像 增强有哪些方法	97
3.5 什么是随机变量	69	4.4 什么是图像的直方图	97
3.6 如何描述随机变量	69	4.5 什么时候有必要修改图像的直方图	97
3.7 什么是事件的概率	69	4.6 如何修改图像的直方图	97
3.8 什么是随机变量的分布函数	70	4.7 什么是直方图均衡化	98
3.9 什么是随机变量取特定值的概率	70	4.8 为什么直方图均衡化并不会产生 具有平坦直方图的图像	98
3.10 什么是随机变量的概率密度函数	71	4.9 增强图像使它具有绝对平坦的 直方图是否可能	98
3.11 如何描述多个随机变量	71	4.10 如果我们不想要一幅有绝对平坦 直方图的图像应该怎么做	100
3.12 n 个随机变量之间的可能关系是 怎样的	71	4.11 为什么除了进行直方图均衡化 之外还希望实现其他的操作	101
3.13 如何定义随机场	72	4.12 如果图像的对比度不均一该如何 处理	102
3.14 如何联系同一随机场中两个随机 变量	73	4.13 对直方图的操作是否还有其他的 方法	103
3.15 如何联系两个随机场中两个随机 变量	73	4.14 如何提高多光谱图像的对比度	105
3.16 既然我们总是只有图像的一个版本, 如何计算以前定义中的期望值	74	4.15 什么是主分量分析	106
3.17 什么时候随机场是齐次的	74	4.16 本章讨论的K-L变换和第3章所讨论 的有什么联系	106
3.18 如何计算随机场的空间统计量	75	4.17 如何实现主分量分析	106
3.19 什么时候随机场是遍历性的	75	4.18 使用主分量来表示图像的优点是 什么	107
3.20 什么时候随机场对于均值是 遍历性的	75	4.19 主分量分析的缺点是什么	107
3.21 什么时候随机场对于自相关函数是 遍历性的	75		

4.20 有一些增强了对比度的图像看起来 噪声很明显。如何处理这种情况	112	5.17 为什么根据Chebyshev范数而不是均 方误差来评价一个近似值	134
4.21 图像噪声的类型有哪些	112	5.18 如何获得一个系统函数的逼近	135
4.22 什么是排序滤波器	114	5.19 什么是窗口法	135
4.23 什么是中值滤波	114	5.20 窗口法存在什么问题	135
4.24 如果图像的噪声不是脉冲的应该 怎么办	114	5.21 如何提高窗口法处理的效果	135
4.25 为什么低通滤波可以减少噪声	115	5.22 如何用一维信号的窗口函数来定义 图像的窗口函数	136
4.26 如果我们感兴趣的是图像的高频 部分应该怎么做	116	5.23 所要解决的逼近问题的形式定义 是什么	136
4.27 什么是理想高通滤波器	116	5.24 什么是线性规划	136
4.28 如何改善有光照变化的图像	116	5.25 如何把滤波器设计问题转化为线性 规划问题	137
4.29 是否可以用第2章的线性方法实现 图像增强的目的	118	5.26 如何减少线性规划求解的计算强度	141
4.30 本章要点	119	5.27 迭代方法的主要思想是什么	141
第5章 二维滤波器	121	5.28 有什么算法可以减少吻合误差的 上限吗	141
5.1 本章概述	121	5.29 最大化算法是如何工作的	141
5.2 如何定义一个二维滤波器	121	5.30 什么是等式的极限集合	142
5.3 系统函数和滤波器的单位采样响应 是如何联系在一起的	121	5.31 什么是La Vallee Poussin定理	142
5.4 为什么我们对实域的滤波器函数 感兴趣	121	5.32 如何证明La Vallee Poussin定理	142
5.5 $h(k, l)$ 需要满足什么条件才可以做为 一个卷积滤波器	122	5.33 迭代算法的步骤是怎样的	142
5.6 一维理想低通滤波器和二维理想低通 滤波器之间的关系	125	5.34 可以逼近一个在频域上有效的 滤波器吗	143
5.7 如何实现无限域上的滤波器	126	5.35 如何构造一个函数去实现用 滤波器的其他频率值表达它 的某些频率的值	143
5.8 如何定义数字化一维滤波器的 z 变换	126	5.36 当仅在频域设计滤波器时要 怎样做	149
5.9 为什么要用 z 变换	127	5.37 如何求解未知值 $H(k, l)$	150
5.10 二维的 z 变换如何定义	127	5.38 根据Chebyshev准则频率采样方法 是否能得到最优解	150
5.11 一维递归滤波器和二维递归滤波器 有什么基本的区别	133	5.39 本章要点	151
5.12 如何知道滤波器没有扩大噪声	134	第6章 图像复原	153
5.13 是否有另一种方法使用无限脉冲 响应滤波器	134	6.1 什么是图像复原	153
5.14 为什么需要逼近理论	134	6.2 图像增强和图像复原之间的区别 是什么	153
5.15 如何知道一个近似滤波器是否性能 良好	134	6.3 为什么图像需要复原	153
5.16 对一个给定的理想系统函数的最佳 逼近是什么	134	6.4 几何畸变是如何产生的	153
		6.5 几何畸变的图像如何被复原	154

6.6 如何实现空间变换	154	6.27 假定退化过程是线性的,为什么 我们要用卷积定理而不是通过解 一组线性方程来恢复结果呢	182
6.7 为什么灰度插值是必要的	154	6.28 式(6-76)看起来简单明了,为什么还 要那么麻烦地去用其他方法呢	183
6.8 退化图像是如何依赖非退化图像和 线性退化过程的点扩展函数的	157	6.29 是否存在对矩阵 H 求逆的方法	184
6.9 退化图像是如何依赖于非退化图像 和线性移不变退化过程的点扩展 函数的	157	6.30 什么矩阵是块循环的	184
6.10 对于离散图像式(6-5)的形式是 怎样的	158	6.31 什么矩阵是循环矩阵	184
6.11 图像复原的问题是什么	158	6.32 为什么块循环矩阵可以很容易地 求逆	184
6.12 如何解决图像复原的问题	158	6.33 什么是循环矩阵的特征值和特征 向量	184
6.13 如何获得退化过程的传递函数 $\hat{H}(u,v)$ 的信息	158	6.34 已知矩阵特征值和特征向量如何 求矩阵的逆	185
6.14 如果知道退化过程的传递函数, 图像复原问题的解决方法是否 就更简单	165	6.35 如何知道表达线性退化过程的矩 阵 H 是块循环的	189
6.15 在 $\hat{H}(u,v) = 0$ 的点 (u,v) 会发生什么 样的情况	165	6.36 如何对角化一个块循环矩阵	190
6.16 是否 $\hat{H}(u,v)$ 和 $\hat{G}(u,v)$ 的零点总是 一致的	165	6.37 现在我们知道了如何解决对 H 求 逆的问题,但是如何解决式(6-76) 对噪声的极端敏感问题	198
6.17 当我们写线性退化方程的时候如何 把噪声考虑进去	165	6.38 如何利用矩阵逆的约束	199
6.18 如何避免扩大噪声	166	6.39 Wiener滤波器和加约束的矩阵逆 滤波器之间的关系是怎样的	201
6.19 如何形式化地表达图像复原问题	171	6.40 本章要点	208
6.20 式(6-37)的解是什么	172	第7章 图像分割和边缘检测	209
6.21 可以求出式(6-37)的线性解吗	172	7.1 本章概述	209
6.22 图像复原问题的线性最小均方误差 解是什么	172	7.2 图像分割和边缘检测的目的是 什么	209
6.23 如果原图像 $f(r)$ 未知,如何利用依 赖于退化图像的互谱密度函数的 式(6-41)来推导需要的滤波器	173	7.3 如何将一幅图像分成一些统一的 区域	209
6.24 如果我们对于未知图像 $f(r)$ 的统计 信息完全不知道,应该如何使用 式(6-47)	174	7.4 给图像加“标记”有什么意义	210
6.25 Wiener滤波器式(6-47)和式(6-25) 的逆滤波器之间的关系是怎样的	174	7.5 当直方图的谷不是很明显时如何 处理	210
6.26 假定知道未知图像 $f(r)$ 的统计信息, 能否确定由 $S_w(r)$ 所表达的噪声的 统计信息	174	7.6 如何最小化错分像素的数目	211
		7.7 如何选择最小误差阈值	211
		7.8 当物体和背景像素是正态分布时的 最小误差阈值是什么	215
		7.9 式(7-6)的两个解的意义是什么	216
		7.10 最小误差阈值方法的缺点是什么	219
		7.11 是否有一种可以不依赖于物体和 背景像素分布模型的方法	219

7.12 Otsu方法有缺点吗	222	7.23 可以选择的最小窗口是什么样的	228
7.13 如何对在变化光照下所获得的图像 定阈值	222	7.24 当图像有噪声的时候会发生什么 情况	229
7.14 如果可以根据 $\ln f(x, y)$ 的直方图来 定图像的阈值, 是否可以根据图像 表面的反射性质来定阈值	222	7.25 对于边缘检测如何选择 3×3 模板的 权重	232
7.15 如果直接阈值化方法在变化光照的 情况下失效, 应该如何处理	224	7.26 参数 K 的最佳值是什么	233
7.16 阈值方法有哪些缺点	225	7.27 一般情况下, 如何决定一个像素 是否是边缘像素	238
7.17 如何处理看起来一致而实际上包含 了不一致区域的图像	226	7.28 Sobel模板是否对所有图像都适用	241
7.18 有哪些方法考虑了像素的空间邻接 关系	226	7.29 如果因为图像中存在着明显的噪声 而需要选择一个更大的模板, 我们 要如何选择权重	241
7.19 如何选择种子像素	226	7.30 能否用边缘检测的最优滤波器以 最优的方式检测图像中的线	243
7.20 拆分和合成方法是如何进行的	227	7.31 阶跃边缘和线的基本差别是什么	244
7.21 与考虑像素之间的相似性相反, 是 否可以考虑区域之间的不相似性来 进行图像分割	227	7.32 本章要点	254
7.22 如何度量相邻像素之间的不相 似性	227	参考文献	255
		主题词索引	259

第1章 导 论

1.1 为什么要做图像处理

图像处理已在三个与图像相关的重要问题上获得了迅速发展:

- 图像数字化和编码, 以便于图像的传输、打印和存储。
- 图像的增强和复原, 以便于更容易地辨别各种探测器所拍摄到的其他行星表面的图像。
- 图像的分割和描述, 可作为机器视觉的前期环节。

1.2 什么是一幅图像

一幅单色图像是一个二维的光强函数 $f(x, y)$, 其中 x 和 y 是空间坐标, f 在 (x, y) 点的值是正比于图像在该点的亮度值的。如果是一幅彩色图像, f 则是一个向量, 它的每一个分量代表图像在该点相应颜色通道(band)的亮度值。

数字图像是在空间坐标和亮度值都离散化的图像 $f(x, y)$, 它可以用一个二维的整数数组来表示, 或者一系列的二维数组来表示, 每一个二维数组代表一个颜色通道。数字化后的亮度值称为灰度级(grey level)的值。

数组的每一个元素称为像素(pixel或pel), 这个名称是来自术语“图像元素”(picture element)。通常, 一般数组的大小是几百乘几百个像素, 并且可能出现大量不同的灰度级。因此, 一个数字化图像通常具有如下的形式:

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,N-1) \end{bmatrix}$$

1

其中 $0 \leq f(x, y) \leq G-1$, 这里 N 和 G 通常用2的整数幂来表示($N = 2^n$, $G = 2^m$)。

1.3 什么是一幅图像在一个像素点的亮度

图像的每一个像素对应着三维世界里的一个实体的一个部分。这个实体受到光线的照射, 其中一些光线被反射, 而另一些被吸收。被反射的一部分光线到达成像传感器, 从而产生了由传感器记录下的这个特定像素点的值。当然, 这个记录下的值是依赖于用来成像的传感器的类型, 以及这种类型的传感器对反射光光谱的响应方式的。但是, 当一个整体场景被同样的传感器成像时, 我们常常会忽略这些细节。值得提醒重要的事就是不同像素的亮度值只是相对而言才有意义, 否则是毫无意义的。因此只有两幅图像被同一个物理过程获取时, 或者已对两幅图像的亮度值进行标准化去除不同物理过程的影响的时候, 我们才比较不同图像之间的像素值。

1.4 为什么图像经常用 512×512 、 256×256 、 128×128 等形式表述

因为当图像的大小是2的次幂时，图像的许多计算可以得到简化。

1.5 存储一幅图像需要多少比特

存储一幅大小为 $N \times N$ ，有 2^m 个不同灰度级的图像所要的比特数 b 是：

$$b = N \times N \times m \quad (1-1)$$

因此，对于一幅典型的 512×512 、有256个灰度级($m = 8$)的图像，需要2 097 152比特或者262 144字节(8比特)。这就是为什么只要不是严重影响图像的质量，我们会尽量减小 m 和 N 的原因。

1.6 什么是图像的分辨率

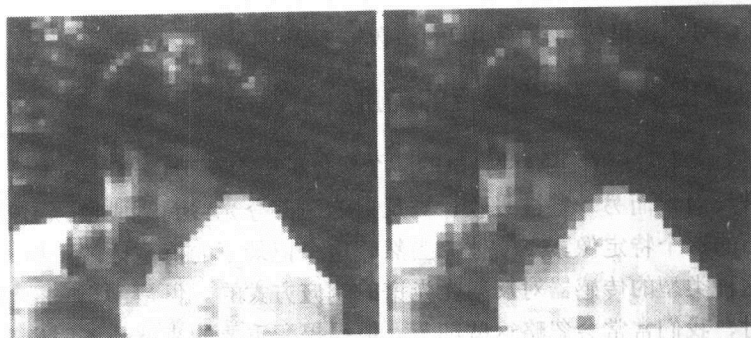
图像的分辨率表示的是能看到图像细节的多少，它显然依赖于 N 和 m 。

保持 m 不变而减少 N 会导致棋盘状效果(checkerboard effect, 见图1-1)。保持 N 不变而减少 m 则会导致假轮廓(false contouring, 见图1-2)。实验表明一幅图像的细节越多，那么用保持 N 恒定而增加 m 的方法来提高图像的显示效果就越不明显。因此，对于一幅细节丰富的图像，比如一幅人群的图像(见图1-3)，所使用的灰度级的数目并不需要太多。



a) 完整大小的图像

b) 原像素数6.17%的图像



c) 原像素数2.78%的图像

d) 原像素数1.54%的图像

图1-1 保持 m 不变而把图像的大小从 338×298 减少到
 84×74 、 56×50 、 42×37 就产生了棋盘状效果

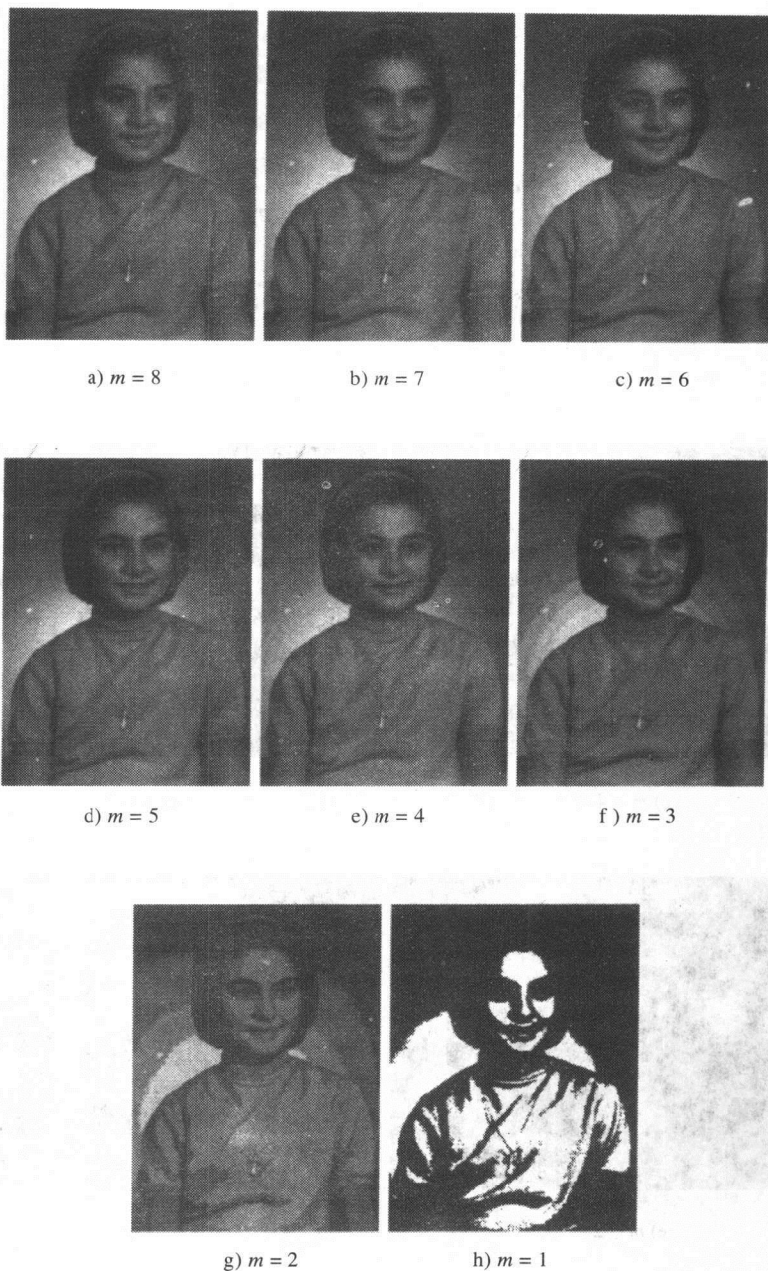


图1-2 保持图像大小不变 (247×333) 而减少灰度级别
($=2^m$) 的数目就产生了假轮廓

1.7 如何进行图像处理

我们用图像变换来进行图像处理, 而图像变换用算子 (operator) 来实现。对一幅输入图像应用一个算子就能产生另一幅图像。本书中我们将主要考虑一类特定的算子, 称为线性算子 (Linear operator)。

a) $m = 8$ b) $m = 6$ c) $m = 4$ d) $m = 3$ e) $m = 2$ f) $m = 1$

图1-3 当保持一幅细节丰富图像的大小不变 (1022×677)
而减少灰度级别的数目时, 结果影响不大

5

1.8 什么是一个线性算子

假定 \mathcal{O} 是一个从图像到图像的算子。如果 f 是一幅图像, $\mathcal{O}(f)$ 是把 \mathcal{O} 应用于 f 后的结果。 \mathcal{O} 是线性的, 如果

$$\mathcal{O}[af + bg] = a\mathcal{O}[f] + b\mathcal{O}[g] \quad (1-2)$$

对所有的图像 f 和 g 和所有的标量 a 和 b 成立。

1.9 算子是如何定义的

算子是用它们的点扩散函数 (point spread function) 来定义的。算子的点扩散函数就是把算子应用于点光源所得到的结果:

$$\mathcal{O}[\text{点光源}] = \text{点扩散函数} \quad (1-3)$$

或者

$$\mathcal{O}[\delta(x-\alpha, y-\beta)] = h(x, \alpha, y, \beta) \quad (1-4)$$

其中, $\delta(x-\alpha, y-\beta)$ 是中心在点 (α, β) 且亮度值为1的点光源。

1.10 一个算子是如何对一幅图像进行变换的

如果算子是线性的, 当点光源的亮度值是原来的 a 倍, 那么结果也将是原来的 a 倍:

$$\mathcal{O}[a\delta(x-\alpha, y-\beta)] = ah(x, \alpha, y, \beta) \quad (1-5)$$

图像可看作具有自身亮度值的点光源 (像素点) 的集合。我们可以说图像是这些点光源的总和。那么用点扩散函数 $h(x, \alpha, y, \beta)$ 来刻划的算子对图像 $f(x, y)$ 作用的效果可以写为:

$$g(\alpha, \beta) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y)h(x, \alpha, y, \beta) \quad (1-6)$$

其中 $g(\alpha, \beta)$ 是输出“图像”, $f(x, y)$ 是输入图像, 并且大小为 $N \times N$ 。

6

1.11 什么是点扩散函数

点扩散函数 $h(x, \alpha, y, \beta)$ 表示位置 (x, y) 的输入值对位置 (α, β) 的输出值影响的大小。如果这种用点扩散函数表示的影响与实际位置不相关, 而是仅依赖于影响和被影响像素之间的相对位置, 我们就有位移不变 (shift invariant) 的点扩散函数:

$$h(x, \alpha, y, \beta) = h(\alpha - x, \beta - y) \quad (1-7)$$

这时, 式(1-6)就是卷积:

$$g(\alpha, \beta) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y)h(\alpha - x, \beta - y) \quad (1-8)$$

如果被影响的列与图像的行是独立的, 那么点扩散函数就是可分离的 (separable):

$$h(x, \alpha, y, \beta) \equiv h_c(x, \alpha)h_r(y, \beta) \quad (1-9)$$

上述表达式也作为函数 $h_c(x, \alpha)$ 和 $h_r(y, \beta)$ 的定义。这时, 式(1-6)可以被写成两个一维变换的叠加:

$$g(\alpha, \beta) = \sum_{x=0}^{N-1} h_c(x, \alpha) \sum_{y=0}^{N-1} f(x, y) h_r(y, \beta) \quad (1-10)$$

如果点扩散函数既是位移不变的且是可分离的, 那么式(1-6)可以被写成两个一维卷积的叠加:

$$g(\alpha, \beta) = \sum_{x=0}^{N-1} h_c(\alpha - x) \sum_{y=0}^{N-1} f(x, y) h_r(\beta - y) \quad (1-11)$$

B1.1 连续域的点光源的形式化定义

定义一个亮度值恒定的扩展光源:

$$\delta_n(x, y) \equiv n^2 \text{rect}(nx, ny) \quad (1-12)$$

其中 n 是正数并且

$$\text{rect}(nx, ny) \equiv \begin{cases} 1 & \text{矩形区域 } |nx| < \frac{1}{2}, |ny| < \frac{1}{2} \text{ 内} \\ 0 & \text{否则} \end{cases} \quad (1-13)$$

该光源的整体亮度值为

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \delta_n(x, y) dx dy = n^2 \underbrace{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \text{rect}(nx, ny) dx dy}_{\text{矩形区域}} = 1 \quad (1-14)$$

它不依赖于 n 。

当 $n \rightarrow \infty$ 时, 产生了一个关于扩展正方形光源的序列 δ_n , 该光源序列在光源大小逐渐收缩时, 能够保持整体亮度值不变。在极限状态下, δ_n 就成为了Dirac delta函数

$$\delta(x, y) \begin{cases} \neq 0 & x = y = 0 \text{ 时} \\ = 0 & \text{其他} \end{cases} \quad (1-15)$$

它有如下性质

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \delta(x, y) dx dy = 1. \quad (1-16)$$

积分

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \delta_n(x, y) g(x, y) dx dy \quad (1-17)$$

是图像 $g(x, y)$ 在一个以 $(0, 0)$ 为中心、以 $1/n$ 为边长的正方形内的平均值。趋于极限时有

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \delta(x, y) g(x, y) dx dy = g(0, 0) \quad (1-18)$$

这是图像在原点的值。类似

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) \delta_n(x-a, y-b) dx dy \quad (1-19)$$

是 g 在中心为 $x=a, y=b$ 边长为 $1/n$ 的正方形内的均值, 因为:

$$\begin{aligned} \delta_n(x-a, y-b) &= n^2 \text{rect}[n(x-a), n(y-b)] \\ &= \begin{cases} n^2 & |n(x-a)| < \frac{1}{2} \quad |n(y-b)| < \frac{1}{2} \\ 0 & \text{其他} \end{cases} \end{aligned} \quad (1-20)$$

可以看出一个中心位于 (a, b) 的正方形光源, 考虑到 $|n(x-a)| < \frac{1}{2}$ 意味着 $-\frac{1}{2} < n(x-a) < \frac{1}{2}$, 也就是 $-\frac{1}{2n} < x-a < \frac{1}{2n}$ 或者 $a - \frac{1}{2n} < x < a + \frac{1}{2n}$ 。因此在区域 $a - \frac{1}{2n} < x < a + \frac{1}{2n}$, $b - \frac{1}{2n} < y < b + \frac{1}{2n}$ 内有 $\delta_n(x-a, y-b) = n^2$ 。

在 $n \rightarrow \infty$ 时, 积分(式(1-19))就等于图像在 $x=a, y=b$ 的值了, 即

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) \delta_n(x-a, y-b) dx dy = g(a, b) \quad (1-21)$$

这个式(1-21)被称为delta函数的平移性质。该式也说明任意图像 $g(x, y)$ 都可以被表示成点光源的叠加。

8

1.12 如何表达一个线性算子作用于一幅图像的实际效果

这需要借助矩阵来实现的。可以按照下式重写式(1-6):

$$\begin{aligned} g(\alpha, \beta) &= f(0, 0)h(0, \alpha, 0, \beta) + f(1, 0)h(1, \alpha, 0, \beta) + \dots + f(N-1, 0)h(N-1, \alpha, 0, \beta) \\ &+ f(0, 1)h(0, \alpha, 1, \beta) + f(1, 1)h(1, \alpha, 1, \beta) + \dots + f(N-1, 1)h(N-1, \alpha, 1, \beta) \\ &+ \dots + f(0, N-1)h(0, \alpha, N-1, \beta) + f(1, N-1)h(1, \alpha, N-1, \beta) + \dots \\ &+ f(N-1, N-1)h(N-1, \alpha, N-1, \beta) \end{aligned} \quad (1-22)$$

表达式的右边可以看做是向量

$$\begin{aligned} \mathbf{h}_{\alpha\beta}^T &\equiv [h(0, \alpha, 0, \beta), h(1, \alpha, 0, \beta), \dots, h(N-1, \alpha, 0, \beta), h(0, \alpha, 1, \beta), h(1, \alpha, 1, \beta), \dots, \\ &h(N-1, \alpha, 1, \beta), \dots, h(1, \alpha, N-1, \beta), \dots, h(N-1, \alpha, N-1, \beta)] \end{aligned} \quad (1-23)$$

和向量

$$\begin{aligned} \mathbf{f}^T &\equiv [f(0, 0), f(1, 0), \dots, f(N-1, 0), f(0, 1), f(1, 1), \dots, f(N-1, 1), \\ &\dots, f(0, N-1), f(1, N-1), \dots, f(N-1, N-1)] \end{aligned} \quad (1-24)$$

的点乘。最后一个向量实际上是图像 $f(x, y)$ 按列组成的向量。如果假定用同样的方式来书写

$g(\alpha, \beta)$, 那么向量 $\mathbf{h}_{\alpha\beta}^T$ 可以看作是矩阵 H 的行, 其中对 $\alpha = 0, \beta$ 从 0 到 $N-1$ 给出了矩阵的第一个 N 行, 然后对 $\alpha = 1, \beta$ 再次从 0 到 $N-1$ 给出了矩阵的第二个 N 行, 依此类推下去。因此, 式(1-6)可以用一种更简洁的方式表达:

$$\boxed{\mathbf{g} = H\mathbf{f}} \quad (1-25)$$

9 这就是线性图像处理的基础方程 (Fundamental Equation)。其中 H 是由 $N \times N$ 个大小均为 $N \times N$ 的子矩阵所构成的 $N^2 \times N^2$ 的方阵, 它按如下方式排列:

$$H = \begin{pmatrix} \alpha \downarrow \left(\begin{matrix} x \rightarrow \\ y=0 \\ \beta=0 \end{matrix} \right) & \alpha \downarrow \left(\begin{matrix} x \rightarrow \\ y=1 \\ \beta=0 \end{matrix} \right) & \cdots & \alpha \downarrow \left(\begin{matrix} x \rightarrow \\ y=N-1 \\ \beta=0 \end{matrix} \right) \\ \alpha \downarrow \left(\begin{matrix} x \rightarrow \\ y=0 \\ \beta=1 \end{matrix} \right) & \alpha \downarrow \left(\begin{matrix} x \rightarrow \\ y=1 \\ \beta=1 \end{matrix} \right) & \cdots & \alpha \downarrow \left(\begin{matrix} x \rightarrow \\ y=N-1 \\ \beta=1 \end{matrix} \right) \\ \vdots & \vdots & & \vdots \\ \alpha \downarrow \left(\begin{matrix} x \rightarrow \\ y=0 \\ \beta=N-1 \end{matrix} \right) & \alpha \downarrow \left(\begin{matrix} x \rightarrow \\ y=1 \\ \beta=N-1 \end{matrix} \right) & \cdots & \alpha \downarrow \left(\begin{matrix} x \rightarrow \\ y=N-1 \\ \beta=N-1 \end{matrix} \right) \end{pmatrix} \quad (1-26)$$

在这种表示法中, 每一个括起来的表达式都代表了一个 $N \times N$ 的子矩阵。每个子矩阵是由函数 $h(x, \alpha, y, \beta)$ 对 y 和 β 取固定的值, 并且对变量 x 和 α 按箭头的方向取遍所有可能的值而构成的。称矩阵 H 的这种模式结构为具有 N^2 个子方阵的矩阵分块(partition), 每个分块具有 N^2 个子方阵。

B1.2 什么是堆积算子?

堆积算子使得我们可以把一个 $N \times N$ 的图像数组写成一个 $N^2 \times 1$ 的向量, 或者把一个 $N^2 \times 1$ 的向量写出一个 $N \times N$ 的方阵数组。

按照如下方式定义一些向量 \mathbf{V}_n 和一些矩阵 N_n :

$$\mathbf{V}_n = \left[\begin{matrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{matrix} \right] \left\{ \begin{array}{l} \text{从 } 0 \text{ 到 } n-1 \text{ 行} \\ \text{第 } n \text{ 行} \\ \text{从行 } n+1 \text{ 到 } N \text{ 行} \end{array} \right.$$

$$N_n = \left[\begin{matrix} & & & & \\ & \mathbf{0} & & & \\ & 1 & 0 & \cdots & 0 \\ & 0 & 1 & \cdots & 0 \\ & \vdots & \vdots & & \vdots \\ & 0 & 0 & \cdots & 1 \\ & & & & \\ & & & & \mathbf{0} \end{matrix} \right] \left\{ \begin{array}{l} \text{纵向依次排列 } n-1 \text{ 个 } N \times N \text{ 的方阵,} \\ \text{它们的每个元素都是 } 0 \\ \text{第 } n \text{ 个矩阵是单位矩阵} \\ \text{纵向依次排列 } N-n \text{ 个 } N \times N \text{ 的方阵,} \\ \text{它们的每个元素都是 } 0 \end{array} \right.$$

\mathbf{V}_n 的维数是 $N \times 1$, N_n 的维数是 $N^2 \times N$, 那么对应于 $N \times N$ 方阵 f 的向量 \mathbf{f} 就可以表示为:

$$\mathbf{f} = \sum_{n=1}^N N_n f \mathbf{V}_n \quad (1-27)$$

可以证明如果 \mathbf{f} 是一个 $N^2 \times 1$ 的向量, 可以把它写出是一个 $N \times N$ 的矩阵 f , 其中, 第一列是由 \mathbf{f} 的前 N 个元素组成, 第二列是由 \mathbf{f} 的第二组 N 个元素组成, 依此类推。用下面的表达式来表示:

$$f = \sum_{n=1}^N N_n^T \mathbf{f} \mathbf{V}_n^T \quad (1-28)$$

• 例1.1(B) 给定一幅 3×3 的图像 f , 要求用堆积算子把图像按向量的形式写出来。

• 假设:

$$f = \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix}$$

对于 $n = 1, 2, 3$, 定义向量 \mathbf{V}_n 和矩阵 N_n :

$$\mathbf{V}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{V}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{V}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$N_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad N_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad N_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

根据式(1-27):

$$\mathbf{f} = N_1 f \mathbf{V}_1 + N_2 f \mathbf{V}_2 + N_3 f \mathbf{V}_3 \quad (1-29)$$

分别计算每一项:

$$N_1 f \mathbf{V}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \Rightarrow$$

$$N_1 f \mathbf{V}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} f_{11} \\ f_{21} \\ f_{31} \end{pmatrix} = \begin{pmatrix} f_{11} \\ f_{21} \\ f_{31} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

类似

$$N_2 f \mathbf{V}_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ f_{12} \\ f_{22} \\ f_{32} \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad N_3 f \mathbf{V}_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ f_{13} \\ f_{23} \\ f_{33} \end{pmatrix} \quad (1-30)$$

然后代入式(1-29)就得到向量 \mathbf{f} 。

12 例1.2(B) 给定一个 9×1 的向量 \mathbf{f} 。使用堆积算子把它写成一个 3×3 的矩阵。

假设:

$$\mathbf{f} = \begin{pmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{pmatrix}$$

根据式(1-28)

$$\mathbf{f} = N_1^T \mathbf{f} \mathbf{V}_1^T + N_2^T \mathbf{f} \mathbf{V}_2^T + N_3^T \mathbf{f} \mathbf{V}_3^T \quad (1-31)$$

分别计算每一项:

$$N_1^T \mathbf{f} \mathbf{V}_1^T = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{pmatrix} (1 \ 0 \ 0)$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} f_{11} & 0 & 0 \\ f_{21} & 0 & 0 \\ f_{31} & 0 & 0 \\ f_{12} & 0 & 0 \\ f_{22} & 0 & 0 \\ f_{32} & 0 & 0 \\ f_{13} & 0 & 0 \\ f_{23} & 0 & 0 \\ f_{33} & 0 & 0 \end{pmatrix} = \begin{pmatrix} f_{11} & 0 & 0 \\ f_{21} & 0 & 0 \\ f_{31} & 0 & 0 \end{pmatrix} \quad (1-32) \quad \boxed{13}$$

类似

$$N_2^T \mathbf{f} \mathbf{V}_2^T = \begin{pmatrix} 0 & f_{12} & 0 \\ 0 & f_{22} & 0 \\ 0 & f_{32} & 0 \end{pmatrix}, \quad N_3^T \mathbf{f} \mathbf{V}_3^T = \begin{pmatrix} 0 & 0 & f_{13} \\ 0 & 0 & f_{23} \\ 0 & 0 & f_{33} \end{pmatrix} \quad (1-33)$$

然后代入式(1-31)就得到矩阵 f 。

...

1.13 矩阵 H 结构的可分离性假设的含义是什么

根据可分离性假设, 可以用 $h_c(x, \alpha)h_r(y, \beta)$ 来代替 $h(x, \alpha, y, \beta)$ 。然后在式(1-26)的 H 每一个分块中, 让 $h_c(x, \alpha)$ 保持常数来重写 H :

$$\begin{pmatrix} h_{r00} \begin{pmatrix} h_{c00} & \dots & h_{cN-10} \\ h_{c01} & \dots & h_{cN-11} \\ \vdots & & \vdots \\ h_{c0N-1} & \dots & h_{cN-1N-1} \end{pmatrix} & \dots & h_{rN-10} \begin{pmatrix} h_{c00} & \dots & h_{cN-10} \\ h_{c01} & \dots & h_{cN-11} \\ \vdots & & \vdots \\ h_{c0N-1} & \dots & h_{cN-1N-1} \end{pmatrix} \\ h_{r01} \begin{pmatrix} h_{c00} & \dots & h_{cN-10} \\ h_{c01} & \dots & h_{cN-11} \\ \vdots & & \vdots \\ h_{c0N-1} & \dots & h_{cN-1N-1} \end{pmatrix} & \dots & h_{rN-11} \begin{pmatrix} h_{c00} & \dots & h_{cN-10} \\ h_{c01} & \dots & h_{cN-11} \\ \vdots & & \vdots \\ h_{c0N-1} & \dots & h_{cN-1N-1} \end{pmatrix} \\ \vdots & & \vdots \\ h_{r0N-1} \begin{pmatrix} h_{c00} & \dots & h_{cN-10} \\ h_{c01} & \dots & h_{cN-11} \\ \vdots & & \vdots \\ h_{c0N-1} & \dots & h_{cN-1N-1} \end{pmatrix} & \dots & h_{rN-1N-1} \begin{pmatrix} h_{c00} & \dots & h_{cN-10} \\ h_{c01} & \dots & h_{cN-11} \\ \vdots & & \vdots \\ h_{c0N-1} & \dots & h_{cN-1N-1} \end{pmatrix} \end{pmatrix}$$

其中, 函数 $h_c(x, \alpha)$ 和 $h_r(y, \beta)$ 的参数被写成存储空间的索引指标。则我们称矩阵 H 是矩阵 h_c^T 和 h_r^T 的Kronecker乘积 (Kronecker product), 并表示为:

$$H = h_c^T \otimes h_r^T \quad (1-34) \quad \boxed{14}$$

1.14 可分离变换如何用矩阵形式来表示

再来考虑表示图像可分离线性变换的式(1-10):

$$g(\alpha, \beta) = \sum_{x=0}^{N-1} h_c(x, \alpha) \sum_{y=0}^{N-1} f(x, y) h_r(y, \beta) \quad (1-35)$$

注意到因子 $\sum_{y=0}^{N-1} f(x, y) h_r(y, \beta)$ 实际上代表了两个 $N \times N$ 矩阵的乘积，它一定是另一个同样大小的矩阵，不妨将它定义为：

$$s(x, \beta) \equiv \sum_{y=0}^{N-1} f(x, y) h_r(y, \beta) = f h_r \quad (1-36)$$

那么式(1-35)可以被写为：

$$g(\alpha, \beta) = \sum_{x=0}^{N-1} h_c(x, \alpha) s(x, \beta) = h_c^T s \quad (1-37)$$

因此用矩阵形式表示为：

$$g = h_c^T f h_r \quad (1-38)$$

1.15 可分离性假设的含义是什么

可分离性假设意味着算子 \mathcal{O} （其点扩散函数 $h(x, \alpha, y, \beta)$ 是可分离的）对图像矩阵 f 的行操作与对它的列操作是独立的。这些独立的算子分别用两个矩阵 h_r 和 h_c 来表示。这就是为什么我们选择下标 r 和 c 来指明这些矩阵（ r = 行， c = 列）的原因。

B1.3 可分离矩阵方程的形式化推导

我们可以按如下方式使用式(1-27)、式(1-28)和式(1-25)：首先用关于 \mathbf{g} 的式(1-28)来表示输出图像 \mathbf{g} ：

$$\mathbf{g} = \sum_{m=1}^N N_m^T \mathbf{g} \mathbf{V}_m^T \quad (1-39)$$

然后根据式(1-25)用 H 和 \mathbf{f} 来表示 \mathbf{g} ，并且用关于 \mathbf{f} 的式(1-27)来代替 \mathbf{f} ：

$$\mathbf{g} = H \sum_{n=1}^N N_n \mathbf{f} \mathbf{V}_n \quad (1-40)$$

把式(1-40)代入式(1-39)，用括号把相关因子括起来就得到：

$$\mathbf{g} = \sum_{m=1}^N \sum_{n=1}^N (N_m^T H N_n) \mathbf{f} (\mathbf{V}_n \mathbf{V}_m^T) \quad (1-41)$$

H 是一个 $N^2 \times N^2$ 矩阵。可以把它看作是由 $N \times N$ 个子矩阵堆积而成。然后可以证明 $N_m^T H N_n$ 就是 H_{mn} 这样的子矩阵。根据可分离性假设, 矩阵 H 是矩阵 h_c 和 h_r 的Kronecker乘积:

$$H = h_c^T \otimes h_r^T \quad (1-42)$$

而分块 H_{mn} 实质上是 $h_r^T(m, n)h_c^T$ 。把它代入式(1-41)就可以得到:

$$\begin{aligned} g &= \sum_{m=1}^N \sum_{n=1}^N \underbrace{h_r^T(m, n)}_{\text{一个标量}} h_c^T f(\mathbf{V}_n \mathbf{V}_m^T) \Rightarrow \\ \Rightarrow g &= h_c^T f \sum_{m=1}^N \sum_{n=1}^N h_r^T(m, n) \mathbf{V}_n \mathbf{V}_m^T \end{aligned} \quad (1-43)$$

乘积 $\mathbf{V}_n \mathbf{V}_m^T$ 是一个仅在位置 n 有非零值的 $(N \times 1)$ 矩阵和一个仅在位置 m 有非零值的 $(N \times 1)$ 矩阵的乘积。因此它是一个仅在位置 (n, m) 有非零值的 $N \times N$ 方阵。

当与 $h_r^T(m, n)$ 相乘时, 它就会置换矩阵 h_r^T 在位置 (n, m) 的元素值并使其他元素全部被置零。把指标 m 和 n 全部遍历一次总和就是 h_r 。因此由式(1-43)可以得到:

$$g = h_c^T f h_r \quad (1-44)$$

- 例1.3(B) 给定一个 9×9 的矩阵 H , 它被划分为9个 3×3 的子矩阵。证明 $N_2^T H N_3$ 是矩阵 H 的分块
- H_{23} , 其中 N_2 和 N_3 是堆积算子的矩阵。

16

假设:

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} & | & h_{14} & h_{15} & h_{16} & | & h_{17} & h_{18} & h_{19} \\ h_{21} & h_{22} & h_{23} & | & h_{24} & h_{25} & h_{26} & | & h_{27} & h_{28} & h_{29} \\ h_{31} & h_{32} & h_{33} & | & h_{34} & h_{35} & h_{36} & | & h_{37} & h_{38} & h_{39} \\ - & - & - & - & - & - & - & - & - & - & - \\ h_{41} & h_{42} & h_{43} & | & h_{44} & h_{45} & h_{46} & | & h_{47} & h_{48} & h_{49} \\ h_{51} & h_{52} & h_{53} & | & h_{54} & h_{55} & h_{56} & | & h_{57} & h_{58} & h_{59} \\ h_{61} & h_{62} & h_{63} & | & h_{64} & h_{65} & h_{66} & | & h_{67} & h_{68} & h_{69} \\ - & - & - & - & - & - & - & - & - & - & - \\ h_{71} & h_{72} & h_{73} & | & h_{74} & h_{75} & h_{76} & | & h_{77} & h_{78} & h_{79} \\ h_{81} & h_{82} & h_{83} & | & h_{84} & h_{85} & h_{86} & | & h_{87} & h_{88} & h_{89} \\ h_{91} & h_{92} & h_{93} & | & h_{94} & h_{95} & h_{96} & | & h_{97} & h_{98} & h_{99} \end{pmatrix}$$

则 H_{23} 子矩阵为:

$$H_{23} = \begin{pmatrix} h_{47} & h_{48} & h_{49} \\ h_{57} & h_{58} & h_{59} \\ h_{67} & h_{68} & h_{69} \end{pmatrix}$$

我们将证明它是用下面的表达式所给出的:

$$\begin{aligned}
 N_2^T H N_3 &= \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{14} & h_{15} & h_{16} & h_{17} & h_{18} & h_{19} \\ h_{21} & h_{22} & h_{23} & h_{24} & h_{25} & h_{26} & h_{27} & h_{28} & h_{29} \\ h_{31} & h_{32} & h_{33} & h_{34} & h_{35} & h_{36} & h_{37} & h_{38} & h_{39} \\ h_{41} & h_{42} & h_{43} & h_{44} & h_{45} & h_{46} & h_{47} & h_{48} & h_{49} \\ h_{51} & h_{52} & h_{53} & h_{54} & h_{55} & h_{56} & h_{57} & h_{58} & h_{59} \\ h_{61} & h_{62} & h_{63} & h_{64} & h_{65} & h_{66} & h_{67} & h_{68} & h_{69} \\ h_{71} & h_{72} & h_{73} & h_{74} & h_{75} & h_{76} & h_{77} & h_{78} & h_{79} \\ h_{81} & h_{82} & h_{83} & h_{84} & h_{85} & h_{86} & h_{87} & h_{88} & h_{89} \\ h_{91} & h_{92} & h_{93} & h_{94} & h_{95} & h_{96} & h_{97} & h_{98} & h_{99} \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} h_{17} & h_{18} & h_{19} \\ h_{27} & h_{28} & h_{29} \\ h_{37} & h_{38} & h_{39} \\ h_{47} & h_{48} & h_{49} \\ h_{57} & h_{58} & h_{59} \\ h_{67} & h_{68} & h_{69} \\ h_{77} & h_{78} & h_{79} \\ h_{87} & h_{88} & h_{89} \\ h_{97} & h_{98} & h_{99} \end{pmatrix} \\
&= \begin{pmatrix} h_{47} & h_{48} & h_{49} \\ h_{57} & h_{58} & h_{59} \\ h_{67} & h_{68} & h_{69} \end{pmatrix} = H_{23}
 \end{aligned}$$

17

1.16 本章要点

基于对图像做操作的算子是线性的和可分离的假设, 变换可以用式(1-45)的形式来表示:

$$g = h_c^T f h_r \quad (1-45)$$

其中 f 和 g 分别是输入图像和输出图像, 而 h_c 和 h_r 是表示算子的点扩散函数的矩阵。

1.17 图像处理的目的是什么

图像处理的目的是为了解决下面4个问题:

- 给定一幅图像 f , 选择矩阵 h_c 和 h_r , 使得根据某种主观标准, 输出图像 g 比 f 更好。这就是图像增强 (Image Enhancement) 问题。
- 给定一幅图像 f , 选择矩阵 h_c 和 h_r , 使得在不会损失太多细节的前提下, g 相对 f 而言可以用更少的比特来表示。这就是图像压缩(Image Compression)的问题。
- 给定一幅图像 g 和 $h(x, \alpha, y, \beta)$ 的一个估计来恢复图像 f 。这就是图像复原(Image Restoration)的问题。
- 给定一幅图像 f , 选择矩阵 h_c 和 h_r , 使得输出图像 g 在某一特征上比 f 突出。这就是图像机器视觉的预备知识(preparation of an image for Automatic Vision)问题。

图1-4和图1-5给出了这些过程的例子。

1.18 本书要点

本书介绍了图像处理的数学基础上所述图像处理的四个主要方面的具体应用。本书将会详细讨论图像增强、图像复原和机器视觉的预备知识的理论。而因为图像压缩本身就可以作为一本书的主题，所以本书只是做了简单地介绍。本书主要讨论线性方法，但是也会介绍一些与图像增强和机器视觉的预备知识相关的非线性技术。

18



05

图1-4 上行：原始图像和它的增强图像。下行：原始图像和用原来比特数的40%来表示的图像

19

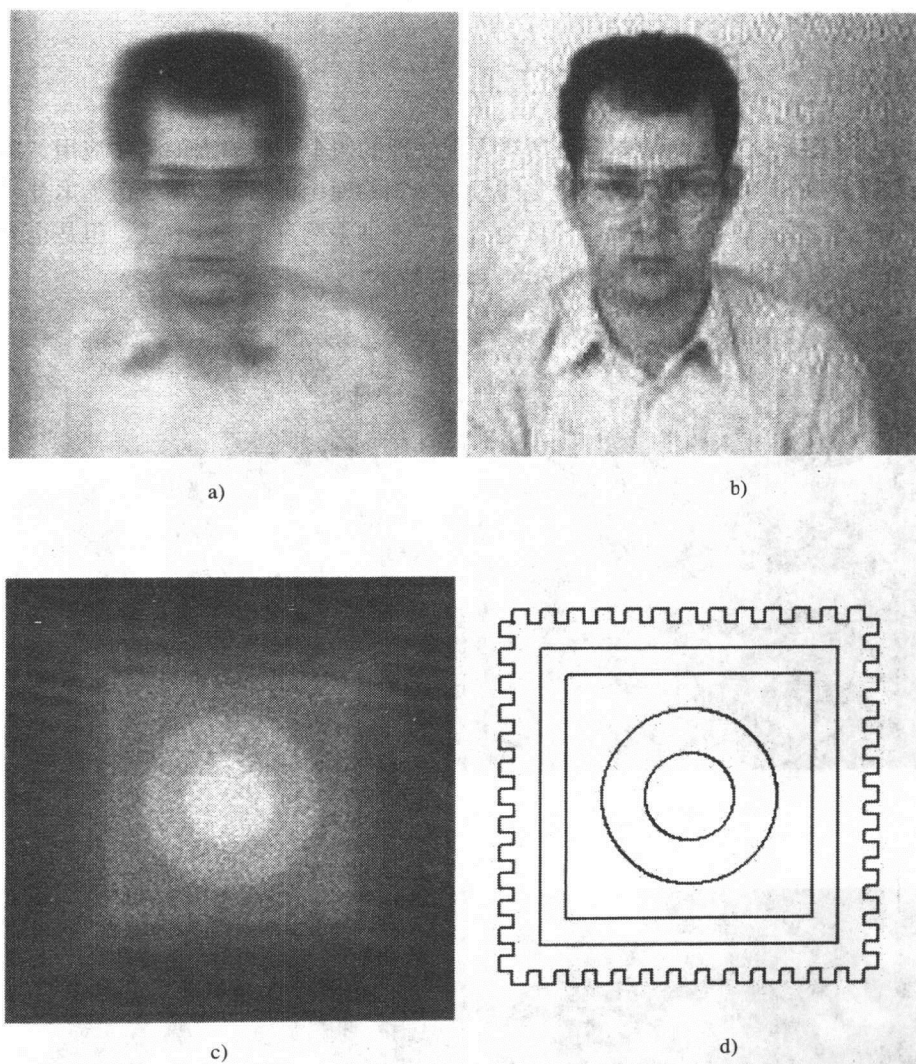


图1-5 上行：模糊了的原始图像和它的复原图像。下行：原始图像和它的边缘图（指示图像亮度值突然改变的位置）

第2章 图像变换

2.1 本章概述

本章涉及了一些最重要的线性图像处理工具，即通过一些基图像的线性叠加来表示图像的方法。

2.2 如何定义基图像

一个基图像可以定义为两个向量的外积 (outer product)。

2.3 什么是两个向量的外积

设有两个 $N \times 1$ 阶的向量：

$$\begin{aligned}\mathbf{u}_i^T &= (u_{i1}, u_{i2}, \dots, u_{iN}) \\ \mathbf{v}_j^T &= (v_{j1}, v_{j2}, \dots, v_{jN})\end{aligned}$$

那么外积定义如下：

$$\mathbf{u}_i \mathbf{v}_j^T = \begin{pmatrix} u_{i1} \\ u_{i2} \\ \vdots \\ u_{iN} \end{pmatrix} \begin{pmatrix} v_{j1} & v_{j2} & \dots & v_{jN} \end{pmatrix} = \begin{pmatrix} u_{i1}v_{j1} & u_{i1}v_{j2} & \dots & u_{i1}v_{jN} \\ u_{i2}v_{j1} & u_{i2}v_{j2} & \dots & u_{i2}v_{jN} \\ \vdots & \vdots & & \vdots \\ u_{iN}v_{j1} & u_{iN}v_{j2} & \dots & u_{iN}v_{jN} \end{pmatrix} \quad (2-1)$$

可见，这两个向量的外积是一个 $N \times N$ 阶矩阵，可以看作是一个图像。

2.4 如何用向量外积进行图像展开

前一章讲过，一个图像矩阵 f 的普通可分离线性变换可表示为：

$$g = h_c^T f h_r \quad (2-2) \quad \boxed{21}$$

其中 g 是输出图像， h_c 和 h_r 是变换矩阵。

可以利用 h_c^T 与 h_r 的逆矩阵以及 g 的适当形式来表示 f ，如下所示：

分别使用 $(h_c^T)^{-1}$ 左乘， h_r^{-1} 右乘上述方程的两端，得

$$(h_c^T)^{-1} g h_r^{-1} = (h_c^T)^{-1} h_c^T f h_r h_r^{-1} = f \quad (2-3)$$

上式可以写成

$$f = (h_c^T)^{-1} g h_r^{-1} \quad (2-4)$$

如果将矩阵 $(h_c^T)^{-1}$ 和 h_r^{-1} 分别按它们的行向量和列向量分块：

$$[\mathbf{u}_1 | \mathbf{u}_2 | \dots | \mathbf{u}_N], \begin{pmatrix} \mathbf{v}_1^T \\ - \\ \mathbf{v}_2^T \\ - \\ \vdots \\ - \\ \mathbf{v}_N^T \end{pmatrix} \quad (2-5)$$

那么有

$$f = (\mathbf{u}_1 \quad \mathbf{u}_2 \quad \dots \quad \mathbf{u}_N) g \begin{pmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \vdots \\ \mathbf{v}_N^T \end{pmatrix} \quad (2-6)$$

矩阵 g 也可以写成 N^2 个 $N \times N$ 阶矩阵之和, 每一个矩阵只有一个非零元素, 如下式所示:

$$g = \begin{pmatrix} g_{11} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} + \begin{pmatrix} 0 & g_{12} & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} + \dots + \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & g_{NN} \end{pmatrix} \quad (2-7)$$

那么式(2-6)可写为

$$f = \sum_{i=1}^N \sum_{j=1}^N g_{ij} \mathbf{u}_i \mathbf{v}_j^T \quad (2-8)$$

这就是图像 f 向量外积的展开形式。外积 $\mathbf{u}_i \mathbf{v}_j^T$ 可看做一幅“图像”, 所以这个和覆盖了外积的所有组合, 用系数 g_{ij} 对其进行适当的加权, 就重构出了原始图像 f 。

22

• 例2.1 导出式(2-8)右边的第 $i=2, j=1$ 项。

• 如果将 g 由式(2-7)代入式(2-6), 那么式(2-6)右边由 N^2 项相似的形式构成, 其中一项为

$$\begin{aligned} & (\mathbf{u}_1 \quad \mathbf{u}_2 \quad \dots \quad \mathbf{u}_N) \begin{pmatrix} 0 & 0 & \dots & 0 \\ g_{21} & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \vdots \\ \mathbf{v}_N^T \end{pmatrix} \\ &= (\mathbf{u}_1 \quad \mathbf{u}_2 \quad \dots \quad \mathbf{u}_N) \begin{pmatrix} 0 & 0 & \dots & 0 \\ g_{21} & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} v_{11} & v_{12} & \dots & v_{1N} \\ v_{21} & v_{22} & \dots & v_{2N} \\ \vdots & \vdots & & \vdots \\ v_{N1} & v_{N2} & \dots & v_{NN} \end{pmatrix} \\ &= \begin{pmatrix} u_{11} & u_{21} & \dots & u_{N1} \\ u_{12} & u_{22} & \dots & u_{N2} \\ \vdots & \vdots & & \vdots \\ u_{1N} & u_{2N} & \dots & u_{NN} \end{pmatrix} \begin{pmatrix} 0 & 0 & \dots & 0 \\ g_{21}v_{11} & g_{21}v_{12} & \dots & g_{21}v_{1N} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} \end{aligned}$$

$$\begin{aligned}
&= \begin{pmatrix} u_{21}g_{21}v_{11} & u_{21}g_{21}v_{12} & \dots & u_{21}g_{21}v_{1N} \\ u_{22}g_{21}v_{11} & u_{22}g_{21}v_{12} & \dots & u_{22}g_{21}v_{1N} \\ \dots & \dots & \dots & \dots \\ u_{2N}g_{21}v_{11} & u_{2N}g_{21}v_{12} & \dots & u_{2N}g_{21}v_{1N} \end{pmatrix} \\
&= g_{21} \begin{pmatrix} u_{21}v_{11} & u_{21}v_{12} & \dots & u_{21}v_{1N} \\ u_{22}v_{11} & u_{22}v_{12} & \dots & u_{22}v_{1N} \\ \dots & \dots & \dots & \dots \\ u_{2N}v_{11} & u_{2N}v_{12} & \dots & u_{2N}v_{1N} \end{pmatrix} = g_{21} \mathbf{u}_2 \mathbf{v}_1^T
\end{aligned}$$

⋮

2.5 什么是酉变换

如果矩阵 h_c 和 h_r 是酉 (unitary) 矩阵, 则式(2-2)表示 f 的一个酉变换 (unitary transform), 而 g 为图像 f 的酉变换域 (unitary transform domain)。

2.6 什么是酉矩阵

如果一个矩阵的逆是它的复共轭转置, 则该矩阵 U 是酉矩阵。也就是说:

$$UU^{T*} = I \quad (2-9) \quad \boxed{23}$$

其中 I 是单位矩阵。通常上标记为 H 而非 T^* 。如果矩阵所有的元素都是实数, 则用正交 (orthogonal)来代替酉。

2.7 什么是酉变换的逆

如果式(2-2)中矩阵 h_c 和 h_r 是酉矩阵, 则它的逆为:

$$f = h_c g h_r^H$$

为了简单起见, 后面用 U 代替 h_c , 用 V 代替 h_r , 则图像 f 的向量外积展开形式可以写成:

$$f = U g V^H \quad (2-10)$$

2.8 如何构造酉矩阵

考察式(2-9)可以得知, 如果矩阵 U 是一个酉矩阵, 则需满足其任意两列的点积必须为0, 而且任意一个列向量的大小必须为1。换句话说, U 是一个酉矩阵, 如果它的列向量构成一组标准正交基。

2.9 如何选择矩阵 U 和 V 使得 g 能用比 f 少的比特来表示

如果要用少于 N^2 个元素表示图像 f , 可以选择矩阵 U 和 V 使得变换后的图像 g 为一个对角矩阵。然后, 结合式(2-8), 仅利用 g 中的 N 个非零元素就可以将 f 表示出来。这可以通过一个称为矩阵对角化 (matrix diagonalization) 的过程来实现, 整个过程被称作图像的奇异值分解 (Singular Value Decomposition, SVD)。

2.10 如何对角化矩阵

可以证明 (见B2.1) 一个秩为 r 的矩阵 g 可以写成如下形式:

$$g = U\Lambda^{\frac{1}{2}}V^T \quad (2-11)$$

24

其中 U 和 V 是 $N \times r$ 阶的正交矩阵, $\Lambda^{\frac{1}{2}}$ 是一个 $r \times r$ 阶的对角矩阵。

- 例2.2 如果 Λ 是一个 2×2 阶的对角矩阵, Λ^m 定义为 Λ 中所有非零元素的 m 次幂组成的矩阵。
- 证明:

$$\Lambda^{-\frac{1}{2}}\Lambda\Lambda^{-\frac{1}{2}} = I \text{ 和 } \Lambda^{-\frac{1}{2}}\Lambda^{\frac{1}{2}} = I.$$

事实上

$$\begin{aligned} \Lambda^{-\frac{1}{2}}\Lambda\Lambda^{-\frac{1}{2}} &= \begin{pmatrix} \lambda_1^{-\frac{1}{2}} & 0 \\ 0 & \lambda_2^{-\frac{1}{2}} \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} \lambda_1^{-\frac{1}{2}} & 0 \\ 0 & \lambda_2^{-\frac{1}{2}} \end{pmatrix} \\ &= \begin{pmatrix} \lambda_1^{-\frac{1}{2}} & 0 \\ 0 & \lambda_2^{-\frac{1}{2}} \end{pmatrix} \begin{pmatrix} \lambda_1^{\frac{1}{2}} & 0 \\ 0 & \lambda_2^{\frac{1}{2}} \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{aligned}$$

这也证明了 $\Lambda^{-\frac{1}{2}}\Lambda^{\frac{1}{2}} = I$ 。

例2.3(B) 假设 H 为一个 3×3 阶的矩阵, 将它分块为一个 2×3 阶的子矩阵 H_1 和一个 1×3 阶的子矩阵 H_2 。证明:

$$H^T H = H_1^T H_1 + H_2^T H_2$$

不妨设

$$H_1 = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \end{pmatrix} \text{ 和 } H_2 = (\tilde{h}_{31} \quad \tilde{h}_{32} \quad \tilde{h}_{33})$$

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ \tilde{h}_{31} & \tilde{h}_{32} & \tilde{h}_{33} \end{pmatrix} \text{ 和 } H^T = \begin{pmatrix} h_{11} & h_{21} & \tilde{h}_{31} \\ h_{12} & h_{22} & \tilde{h}_{32} \\ h_{13} & h_{23} & \tilde{h}_{33} \end{pmatrix} = (H_1^T \mid H_2^T)$$

则

$$H^T H = \begin{pmatrix} h_{11} & h_{21} & \tilde{h}_{31} \\ h_{12} & h_{22} & \tilde{h}_{32} \\ h_{13} & h_{23} & \tilde{h}_{33} \end{pmatrix} \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ \tilde{h}_{31} & \tilde{h}_{32} & \tilde{h}_{33} \end{pmatrix} =$$

25

$$\begin{pmatrix} h_{11}^2 + h_{21}^2 + \tilde{h}_{31}^2 & h_{11}h_{12} + h_{21}h_{22} + \tilde{h}_{31}\tilde{h}_{32} & h_{11}h_{13} + h_{21}h_{23} + \tilde{h}_{31}\tilde{h}_{33} \\ h_{12}h_{11} + h_{22}h_{21} + \tilde{h}_{32}\tilde{h}_{31} & h_{12}^2 + h_{22}^2 + \tilde{h}_{32}^2 & h_{12}h_{13} + h_{22}h_{23} + \tilde{h}_{32}\tilde{h}_{33} \\ h_{13}h_{11} + h_{23}h_{21} + \tilde{h}_{33}\tilde{h}_{31} & h_{13}h_{12} + h_{23}h_{22} + \tilde{h}_{33}\tilde{h}_{32} & h_{13}^2 + h_{23}^2 + \tilde{h}_{33}^2 \end{pmatrix}$$

则可以通过明确的计算, 证明它等于 $H_1^T H_1 + H_2^T H_2$

$$\begin{aligned} H_1^T H_1 &= \begin{pmatrix} h_{11} & h_{21} \\ h_{12} & h_{22} \\ h_{13} & h_{23} \end{pmatrix} \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \end{pmatrix} \\ &= \begin{pmatrix} h_{11}^2 + h_{21}^2 & h_{11}h_{12} + h_{21}h_{22} & h_{11}h_{13} + h_{21}h_{23} \\ h_{12}h_{11} + h_{22}h_{21} & h_{12}^2 + h_{22}^2 & h_{12}h_{13} + h_{22}h_{23} \\ h_{13}h_{11} + h_{23}h_{21} & h_{13}h_{12} + h_{23}h_{22} & h_{13}^2 + h_{23}^2 \end{pmatrix} \\ H_2^T H_2 &= \begin{pmatrix} \tilde{h}_{31} \\ \tilde{h}_{32} \\ \tilde{h}_{33} \end{pmatrix} \begin{pmatrix} \tilde{h}_{31} & \tilde{h}_{32} & \tilde{h}_{33} \end{pmatrix} = \begin{pmatrix} \tilde{h}_{31}^2 & \tilde{h}_{31}\tilde{h}_{32} & \tilde{h}_{31}\tilde{h}_{33} \\ \tilde{h}_{32}\tilde{h}_{31} & \tilde{h}_{32}^2 & \tilde{h}_{32}\tilde{h}_{33} \\ \tilde{h}_{33}\tilde{h}_{31} & \tilde{h}_{33}\tilde{h}_{32} & \tilde{h}_{33}^2 \end{pmatrix} \end{aligned}$$

将 $H_1^T H_1$ 与 $H_2^T H_2$ 相加, 就可以得到与前面相同的结果。

例2.4 已知一个图像用矩阵 g 来表示的。证明: 矩阵 gg^T 是一个对称矩阵。

如果一个矩阵等于其转置, 则该矩阵为对称矩阵。考察 gg^T 的转置:

$$(gg^T)^T = (g^T)^T g^T = gg^T$$

例2.5(B) 证明: 如果将一个 $N \times N$ 阶的矩阵 S 分块成一个 $r \times N$ 阶的子矩阵 S_1 和一个 $(N-r) \times N$ 阶的子矩阵 S_2 , 则有下式成立:

$$SAS^T = \left(\begin{array}{c|c} S_1AS_1^T & S_1AS_2^T \\ \hline S_2AS_1^T & S_2AS_2^T \end{array} \right)$$

26

其中 A 为一个 $N \times N$ 阶矩阵。

$$SAS^T = \begin{pmatrix} S_1 \\ \hline S_2 \end{pmatrix} A (S_1^T \mid S_2^T)$$

考虑矩阵 A 与 $(S_1^T \mid S_2^T)$ 相乘, 用 A 的行与 $(S_1^T \mid S_2^T)$ 的列相乘, 如下图所示:

$$\begin{pmatrix} \cdots & \cdots \\ \cdots & \cdots \\ \cdots & \cdots \end{pmatrix} \begin{pmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

那么很明显, 结果为 $(AS_1^T \mid AS_2^T)$ 。下面考虑 $\begin{pmatrix} S_1 \\ \hline S_2 \end{pmatrix}$ 与 $(AS_1^T \mid AS_2^T)$ 相乘, 用 $\begin{pmatrix} S_1 \\ \hline S_2 \end{pmatrix}$

的行与 $(AS_1^T \mid AS_2^T)$ 的列相乘, 如下图所示:

$$\begin{pmatrix} \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix} \begin{pmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

那么结果显然成立。

例2.6(B) 证明: 如果 $Ag g^T A^T = 0$, 那么 $Ag = 0$, 其中 A 和 g 分别是 $r \times N$ 阶和 $N \times N$ 阶矩阵。

由已知可得

$$Ag g^T A^T = Ag (Ag)^T = 0$$

[27] 其中 Ag 是 $r \times N$ 阶矩阵, 将其记为 B , 则有 $BB^T = 0$:

$$\begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1r} \\ b_{21} & b_{22} & \cdots & b_{2r} \\ \cdots & \cdots & \cdots & \cdots \\ b_{r1} & b_{r2} & \cdots & b_{rr} \end{pmatrix} \begin{pmatrix} b_{11} & b_{21} & \cdots & b_{r1} \\ b_{12} & b_{22} & \cdots & b_{r2} \\ \cdots & \cdots & \cdots & \cdots \\ b_{1r} & b_{2r} & \cdots & b_{rr} \end{pmatrix} = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \Rightarrow$$

$$\begin{pmatrix} b_{11}^2 + b_{12}^2 + \cdots + b_{1r}^2 & \cdots & \cdots & \cdots \\ \cdots & b_{21}^2 + b_{22}^2 + \cdots + b_{2r}^2 & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & b_{r1}^2 + b_{r2}^2 + \cdots + b_{rr}^2 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

根据矩阵相等则对应元素相等, 可以得到

$$b_{11}^2 + b_{12}^2 + \cdots + b_{1r}^2 = 0$$

r 个实数的平方和等于 0, 当且仅当每个实数都为 0。类似地, 对于 BB^T 对角线上的其他元素也是如此, 那么结论得证。

B2.1 任意图像都可以对角化吗?

是。考虑图像 g 和它的转置 g^T 。 gg^T 是一个实对称矩阵 (见例 2.4), 设它有 r 个非零特征向量, λ_i 是它的第 i 个特征值。那么存在一个正交矩阵 S (由 gg^T 的特征向量组成) 使得:

$$\begin{aligned}
 Sgg^TS^T &= \begin{pmatrix} \lambda_1 & 0 & \dots & 0 & | & 0 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 & | & 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots & | & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \lambda_r & | & 0 & 0 & \dots & 0 \\ - & - & - & - & - & - & - & - & - \\ 0 & 0 & \dots & 0 & | & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & | & 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots & | & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 & | & 0 & 0 & \dots & 0 \end{pmatrix} \\
 &= \begin{pmatrix} \Lambda & | & 0 \\ - & - & - \\ 0 & | & 0 \end{pmatrix} \quad (2-12)
 \end{aligned}$$

28

成立。

其中, Λ 与 0 表示上面对角矩阵的子块。类似地, 可以将矩阵 S 分成一个 $r \times N$ 阶矩阵 S_1 与一个 $(N-r) \times N$ 阶矩阵 S_2 :

$$S = \begin{pmatrix} S_1 \\ - \\ S_2 \end{pmatrix}$$

因为 S 是正交的, 利用例 2.3 的结果, 则有

$$\begin{aligned}
 S^TS &= I \Rightarrow S_1^TS_1 + S_2^TS_2 = I \Rightarrow \\
 S_1^TS_1 &= I - S_2^TS_2 \Rightarrow S_1^TS_1g = g - S_2^TS_2g \quad (2-13)
 \end{aligned}$$

由式(2-12)以及例 2.5 和例 2.6, 显然可以得到

$$S_1gg^TS_1^T = \Lambda \quad (2-14)$$

$$S_2gg^TS_2^T = 0 \Rightarrow S_2g = 0 \quad (2-15)$$

把式(2-15)代入式(2-13)可以得到

$$S_1^TS_1g = g \quad (2-16)$$

也就是有 $S_1^TS_1 = I$ 成立, 即 S_1 是一个正交矩阵。将式(2-14)的左右两端同乘以 $\Lambda^{-\frac{1}{2}}$, 得

$$\Lambda^{-\frac{1}{2}}S_1gg^TS_1^T\Lambda^{-\frac{1}{2}} = \Lambda^{-\frac{1}{2}}\Lambda\Lambda^{-\frac{1}{2}} = I \quad (2-17)$$

由于 $\Lambda^{-\frac{1}{2}}$ 是对角阵, $\Lambda^{-\frac{1}{2}} = (\Lambda^{-\frac{1}{2}})^T$ 。所以上述方程可写成

$$\Lambda^{-\frac{1}{2}}S_1g(\Lambda^{-\frac{1}{2}}S_1g)^T = I \quad (2-18)$$

因此, 存在一个矩阵 $q \equiv \Lambda^{-\frac{1}{2}}S_1g$, 它的逆是它的转置 (即是正交的)。可以将矩阵 S_1g 用 $\Lambda^{\frac{1}{2}}q$ 来表示, 并代入式(2-16)得

$$S_1^T\Lambda^{\frac{1}{2}}q = g \quad \text{或} \quad g = S_1^T\Lambda^{\frac{1}{2}}q \quad (2-19)$$

29

换句话说, g 可以用由 gg^T 的非零特征值的平方根组成的对角矩阵 $\Lambda^{\frac{1}{2}}$, 左右分别乘以正交矩阵 S 和 q 表示出来。这个结果表示图像 g 的对角化。

2.11 如何计算图像对角化时所需要的矩阵 U 、 V 及 $\Lambda^{\frac{1}{2}}$

如果对式(2-11)取转置, 则有

$$g^T = V\Lambda^{\frac{1}{2}}U^T \quad (2-20)$$

用式(2-20)乘以式(2-11)得到

$$gg^T = U\Lambda^{\frac{1}{2}}V^TV\Lambda^{\frac{1}{2}}U^T = U\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}U^T \Rightarrow gg^T = U\Lambda U^T \quad (2-21)$$

这表明矩阵 Λ 是由矩阵 gg^T 的 r 个非零特征值组成的, 而 U 是由此矩阵的特征向量组成的。类似地, 如果用式(2-11)乘以式(2-20)得

$$g^Tg = V\Lambda V^T \quad (2-22)$$

这表明矩阵 V 是由矩阵 g^Tg 的特征向量所组成的。

B2.2 如果矩阵 gg^T 的特征值是负的将会怎样?

下面证明 gg^T 的特征值总是非负的数。假设 λ 是矩阵 gg^T 的一个特征值, u 是与之相对应的特征向量, 那么有:

$$gg^T u = \lambda u$$

用 u^T 左乘方程的两端, 有

$$u^T gg^T u = u^T \lambda u$$

λ 是一个数, 因而可以改变它在方程右边的位置, 又根据矩阵乘法的结合性, 则可以写成

$$(u^T g)(g^T u) = \lambda u^T u$$

由于 u 是一个特征向量, 则 $u^T u = 1$ 。因此:

$$(g^T u)^T (g^T u) = \lambda$$

其中 $g^T u$ 是某个向量 y 。则有 $\lambda = y^T y > 0$, 因为 $y^T y$ 的大小等于向量 y 的长度的平方。

30

- 例2.7 如果 λ_i 是 gg^T 的特征值, u_i 是其对应的特征向量, 证明 g^Tg 具有相同的特征值且对应特征向量由 $v_i = g^T u_i$ 得出。

根据定义

$$gg^T u_i = \lambda_i u_i$$

两端都左乘 g^T :

$$g^T g g^T \mathbf{u}_i = g^T \lambda_i \mathbf{u}_i$$

由于 λ_i 是一个标量, 所以可改变它的位置, 又根据矩阵乘法的结合性则有:

$$g^T g (g^T \mathbf{u}_i) = \lambda_i (g^T \mathbf{u}_i)$$

这就证明了 $g^T \mathbf{u}_i$ 是 $g^T g$ 的特征向量, 且相应特征值为 λ_i 。

例2.8 已知一个图像 $g = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$ 。计算 $g g^T$ 的特征向量 \mathbf{u}_i , $g^T g$ 的特征向量 \mathbf{v}_i 。

$$g^T = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

则

$$g g^T = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 6 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

$g g^T$ 的特征值可以由下列特征方程计算得到

$$\begin{vmatrix} 1-\lambda & 2 & 0 \\ 2 & 6-\lambda & 1 \\ 0 & 1 & 1-\lambda \end{vmatrix} = 0 \Rightarrow (1-\lambda)[(6-\lambda)(1-\lambda)-1]-2[2(1-\lambda)]=0$$

$$\Rightarrow (1-\lambda)[(6-\lambda)(1-\lambda)-1-4]=0$$

31

其中一个特征值为 $\lambda=1$, 另外两个为下式的平方根:

$$6-6\lambda-\lambda+\lambda^2-5=0 \Rightarrow \lambda^2-7\lambda+1=0 \Rightarrow \lambda = \frac{7 \pm \sqrt{49-4}}{2} = \frac{7 \pm 6.7}{2}$$

$\lambda = 6.85$ 或 $\lambda = 0.146$

特征值按降幂排列为

$$\lambda_1 = 6.85, \lambda_2 = 1, \lambda_3 = 0.146$$

设 $\mathbf{u}_i = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$ 为对应于特征值 λ_i 的特征向量, 则有

$$\begin{pmatrix} 1 & 2 & 0 \\ 2 & 6 & 1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \lambda_i \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \Rightarrow \begin{cases} x_1 + 2x_2 = \lambda_i x_1 \\ 2x_1 + 6x_2 + x_3 = \lambda_i x_2 \\ x_2 + x_3 = \lambda_i x_3 \end{cases}$$

当 $\lambda_i = 6.85$ 时

$$2x_2 - 5.85x_1 = 0 \quad (2-23)$$

$$2x_1 - 0.85x_2 + x_3 = 0 \quad (2-24)$$

$$x_2 - 5.85x_3 = 0 \quad (2-25)$$

将式(2-24)乘以5.85, 与式(2-25)相加得

$$11.7x_1 - 4x_2 = 0 \quad (2-26)$$

式(2-26)与式(2-23)是相同的, 所以对于这三个未知数, 只有两个独立的方程。不妨设 x_1 的值为1, 那么 $x_2 = 2.927$, 且由式(2-24), $x_3 = -2 + 0.85 \times 2.925 = -2 + 2.5 = 0.5$

因此, 第一个特征向量为

$$\begin{pmatrix} 1 \\ 2.927 \\ 0.5 \end{pmatrix}$$

通过标准化, 也就是除以 $\sqrt{1^2 + 2.927^2 + 0.5^2} = 3.133$, 得

$$\mathbf{u}_1 = \begin{pmatrix} 0.319 \\ 0.934 \\ 0.160 \end{pmatrix}$$

32

当 $\lambda_i = 1$ 时, 所要解的线性方程组为

$$\begin{aligned} x_1 + 2x_2 &= x_1 \Rightarrow x_2 = 0 \\ 2x_1 + x_3 &= 0 \Rightarrow x_3 = -2x_1 \end{aligned}$$

设 $x_1 = 1$, 那么 $x_3 = -2$ 。又因为 $x_2 = 0$, 所以须将每个元素都除以 $\sqrt{1^2 + 2^2} = \sqrt{5}$ 以使得特征向量具有单位长度:

$$\mathbf{u}_2 = \begin{pmatrix} 0.447 \\ 0 \\ -0.894 \end{pmatrix}$$

当 $\lambda_i = 0.146$ 时, 所要解的线性方程组为

$$\begin{aligned} 0.854x_1 + 2x_2 &= 0 \\ 2x_1 + 5.854x_2 + x_3 &= 0 \\ x_2 + 0.854x_3 &= 0 \end{aligned}$$

设 $x_1 = 1$, 那么 $x_2 = -\frac{0.854}{2} = -0.427$, $x_3 = -\frac{0.427}{0.854} = 0.5$ 。因此, 第三个特征向量为

$$\begin{pmatrix} 1 \\ -0.427 \\ 0.5 \end{pmatrix}$$

除以 $\sqrt{1 + 0.427^2 + 0.5^2} = 1.197$, 得

$$\mathbf{u}_3 = \begin{pmatrix} 0.835 \\ -0.357 \\ 0.418 \end{pmatrix}$$

所以 $g^T g$ 的特征向量可由 $g^T \mathbf{u}_i$ 得出; 也就是说, 第一个为

$$\begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0.319 \\ 0.934 \\ 0.160 \end{pmatrix} = \begin{pmatrix} 2.187 \\ 0.934 \\ 1.094 \end{pmatrix}$$

通过除以 $\sqrt{2.187^2 + 0.934^2 + 1.094^2} = 2.618$, 将其标准化为

$$\mathbf{v}_1 = \begin{pmatrix} 0.835 \\ 0.357 \\ 0.418 \end{pmatrix}$$

类似地有

$$\mathbf{v}_2 = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0.447 \\ 0 \\ -0.894 \end{pmatrix} = \begin{pmatrix} 0.447 \\ 0 \\ -0.894 \end{pmatrix}$$

33

而第三个特征向量为

$$\begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0.835 \\ -0.357 \\ 0.418 \end{pmatrix} = \begin{pmatrix} 0.121 \\ -0.357 \\ 0.061 \end{pmatrix}$$

标准化后为

$$\mathbf{v}_3 = \begin{pmatrix} 0.319 \\ -0.934 \\ 0.160 \end{pmatrix}$$

...

2.12 什么是图像的奇异值分解

图像 g 的奇异值分解(SVD)就是指它的向量的外积展开, 其中所用的向量是 gg^T 和 $g^T g$ 的特征向量, 展开系数是这些矩阵的特征值。用图像 g 代替图像 f , 这样式(2-8)可以改写成:

$$g = \sum_{i=1}^r \lambda_i^{\frac{1}{2}} \mathbf{u}_i \mathbf{v}_i^T \quad (2-27)$$

因为惟一的非零项只有当 $i = j$ 时才取得。

2.13 如何用SVD逼近一幅图像

如果在式(2-27)中, 只想保留 $k < r$ 的项, 那么可以重构出图像的近似形式如下:

$$g_k = \sum_{i=1}^k \lambda_i^{\frac{1}{2}} \mathbf{u}_i \mathbf{v}_i^T \quad (2-28)$$

- 例2.9 有一幅具有256个灰度级的大小为 256×256 的图像需要传输。如果传输变换后的图像
 • 比传输原图像效率更低, 则在原图像的奇异值分解中最多有多少项? (假设每个实数需要32比特。)

34 假设将 $\lambda_i^{\frac{1}{2}}$ 和与之对应的向量看作一项, 那么传输图像的SVD展开式中的一项需要传输

$$2 \times 32 \times 256 \text{ 比特。}$$

这是因为我们要传输两个向量, 对一幅 256×256 的图像来说, 每一个向量有256个分量, 又因为都是实数所以每个分量需要32比特。如果要传输整幅图像, 就不得不传输 $256 \times 256 \times 8$ 比特 (因为每个像素需8比特)。因此使SVD传输更有效率, 最大传输项数应不超过:

$$k = \frac{256 \times 256 \times 8}{2 \times 32 \times 256} = \frac{256}{8} = 32$$

•
•
•

2.14 用SVD逼近图像的误差是多少

原图像与近似图像的差为

$$D \equiv g - g_k = \sum_{i=k+1}^r \lambda_i^{\frac{1}{2}} \mathbf{u}_i \mathbf{v}_i^T \quad (2-29)$$

可以通过计算矩阵 D 的范数来计算误差的大小, 也就是计算该矩阵元素的平方和。由式(2-29)可知 D 的 mn 元素为

$$\begin{aligned} d_{mn} &= \sum_{i=k+1}^r \lambda_i^{\frac{1}{2}} u_{im} v_{in} \Rightarrow \\ d_{mn}^2 &= \left(\sum_{i=k+1}^r \lambda_i^{\frac{1}{2}} u_{im} v_{in} \right)^2 \\ &= \sum_{i=k+1}^r \lambda_i u_{im}^2 v_{in}^2 + 2 \sum_{i=k+1}^r \sum_{j=k+1, j \neq i}^r \lambda_i^{\frac{1}{2}} \lambda_j^{\frac{1}{2}} u_{im} v_{in} u_{jm} v_{jn} \\ \Rightarrow \|D\| &= \sum_m \sum_n d_{mn}^2 \\ &= \sum_m \sum_n \sum_{i=k+1}^r \lambda_i u_{im}^2 v_{in}^2 + 2 \sum_m \sum_n \sum_{i=k+1}^r \sum_{j=k+1, j \neq i}^r \lambda_i^{\frac{1}{2}} \lambda_j^{\frac{1}{2}} u_{im} v_{in} u_{jm} v_{jn} \\ &= \sum_{i=k+1}^r \lambda_i \sum_m u_{im}^2 \sum_n v_{in}^2 + 2 \sum_{i=k+1}^r \sum_{j=k+1, j \neq i}^r \lambda_i^{\frac{1}{2}} \lambda_j^{\frac{1}{2}} \sum_m u_{im} u_{jm} \sum_n v_{in} v_{jn} \end{aligned}$$

而 \mathbf{u}_i , \mathbf{v}_i 是特征向量, 因此它们构成一组正交基, 则有

$$\sum_m u_{im}^2 = 1, \quad \sum_n v_{in}^2 = 1$$

35

当 $i \neq j$ 时, 有 $\sum_n v_{in} v_{jn} = 0$ 和 $\sum_m u_{im} u_{jm} = 0$ 。

由于 $i \neq j$ 时, $u_i u_j^T = 0$ 且 $v_i v_j^T = 0$, 则有

$$\|D\| = \sum_{i=k+1}^r \lambda_i \quad (2-30)$$

因此, 利用式(2-28)近似重构图像的误差等于忽略掉的特征值的和。

- 例2.10 对于一个 3×3 的矩阵 D , 证明它的范数(定义为矩阵 $D^T D$ 的迹)等于它的所有元素的平方和。

设

$$D \equiv \begin{pmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{pmatrix}$$

那么

$$D^T D = \begin{pmatrix} d_{11} & d_{21} & d_{31} \\ d_{12} & d_{22} & d_{32} \\ d_{13} & d_{23} & d_{33} \end{pmatrix} \begin{pmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{pmatrix} =$$

$$\begin{pmatrix} d_{11}^2 + d_{21}^2 + d_{31}^2 & d_{11}d_{12} + d_{21}d_{22} + d_{31}d_{32} & d_{31}d_{13} + d_{32}d_{23} + d_{31}d_{33} \\ d_{12}d_{11} + d_{22}d_{21} + d_{32}d_{31} & d_{12}^2 + d_{22}^2 + d_{32}^2 & d_{12}d_{13} + d_{22}d_{23} + d_{32}d_{33} \\ d_{13}d_{11} + d_{23}d_{21} + d_{33}d_{31} & d_{13}d_{12} + d_{23}d_{22} + d_{33}d_{32} & d_{13}^2 + d_{23}^2 + d_{33}^2 \end{pmatrix}$$

$$\text{trace}[D^T D] = (d_{11}^2 + d_{21}^2 + d_{31}^2) + (d_{12}^2 + d_{22}^2 + d_{32}^2) + (d_{13}^2 + d_{23}^2 + d_{33}^2)$$

= 所有项的平方和

•
•
•

2.15 如何最小化重构误差

如果将特征值 λ_i 按降序排列, 截取展开式中的某整数 $k < r$ 项, 则 g_k 为图像 g 的最小平方逼近。这是因为差矩阵元素的平方和最小, 其等于那些未用的特征值的和, 这些特征值是选出的最小值。

36

注意, 这种图像的奇异值分解在最小平方误差的意义下是最优的, 但在对某个图像展开时所用到的基图像(特征图像)时, 由这个图像自身确定(它们由矩阵 $g^T g$ 和 $g g^T$ 的特征向量来决定)。

- 例2.11 在例2.8图像的奇异值分解中只保留第一项而令其他的为0, 验证重构图像的平方误差等于忽略掉的特征值的和。

如果仅保留第一个特征值, 图像则可仅由第一个特征图像来逼近, 用相应特征值的平方根对其进行加权:

$$\begin{aligned}
 g_1 &= \sqrt{\lambda_1} \mathbf{u}_1 \mathbf{v}_1^T = \sqrt{6.85} \begin{pmatrix} 0.319 \\ 0.934 \\ 0.160 \end{pmatrix} \begin{pmatrix} 0.835 & 0.357 & 0.418 \end{pmatrix} \\
 &= \begin{pmatrix} 0.835 \\ 2.444 \\ 0.419 \end{pmatrix} \begin{pmatrix} 0.835 & 0.357 & 0.418 \end{pmatrix} = \begin{pmatrix} 0.697 & 0.298 & 0.349 \\ 2.041 & 0.873 & 1.022 \\ 0.350 & 0.150 & 0.175 \end{pmatrix}
 \end{aligned}$$

则重构误差由 g_1 与原始图像的差得出:

$$g - g_1 = \begin{pmatrix} 0.303 & -0.298 & -0.349 \\ -0.041 & 0.127 & -0.022 \\ -0.350 & -0.150 & 0.825 \end{pmatrix}$$

误差的平方和为:

$$\begin{aligned}
 0.303^2 + 0.298^2 + 0.349^2 + 0.041^2 + 0.127^2 + 0.022^2 + 0.350^2 + 0.150^2 + 0.825^2 \\
 = 1.146
 \end{aligned}$$

这恰好等于两个忽略掉的特征值 λ_2 和 λ_3 的和。

•
•
•

2.16 用SVD进行图像展开时的基图像是怎样的

对于这个问题没有一个明确的答案, 因为这些基图像依赖于每一个被展开的图像本身,

37 它们是其特征图像。

• 例2.12 对下列图像进行奇异值分解(SVD):

$$g = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

以确定上述图像的特征图像。

$$gg^T = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 2 \\ 0 & 1 & 0 \\ 2 & 0 & 2 \end{pmatrix}$$

gg^T 的特征值是下面特征方程的解:

$$\begin{vmatrix} 2-\lambda & 0 & 2 \\ 0 & 1-\lambda & 0 \\ 2 & 0 & 2-\lambda \end{vmatrix} = 0 \Rightarrow (2-\lambda)^2(1-\lambda) - 4(1-\lambda) = 0 \Rightarrow \\
 (1-\lambda)(\lambda-4)\lambda = 0$$

特征值为: $\lambda_1 = 4, \lambda_2 = 1, \lambda_3 = 0$ 。相应的特征向量为

$$\begin{pmatrix} 2 & 0 & 2 \\ 0 & 1 & 0 \\ 2 & 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = 4 \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \Rightarrow \begin{matrix} 2x_1 + 2x_3 = 4x_1 \\ x_2 = 4x_2 \\ 2x_1 + 2x_3 = 4x_3 \end{matrix} \Rightarrow \begin{matrix} x_1 = x_3 \\ x_2 = 0 \end{matrix}$$

令 $x_1 = x_3 = \frac{1}{\sqrt{2}}$ 以使特征向量具有单位长度。这样 $\mathbf{u}_1^T = \left(\frac{1}{\sqrt{2}} \ 0 \ \frac{1}{\sqrt{2}} \right)$

$$\begin{pmatrix} 2 & 0 & 2 \\ 0 & 1 & 0 \\ 2 & 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \Rightarrow \begin{array}{l} 2x_1 + 2x_3 = x_1 \\ x_2 = x_2 \\ 2x_1 + 2x_3 = x_3 \end{array} \left| \begin{array}{l} x_1 = -2x_3 \\ x_2 = x_2 \\ x_3 = -2x_1 \end{array} \right.$$

上面第二个等式不代表任何信息, 可以令 x_2 为任意的值。但是不能令 $x_2 = 0$, 因为第一个等式和第三个等式不能同时成立, 除非 $x_1 = x_3 = 0$ 。如果 x_2 也为 0, 就会得出不代表特征向量的平凡解。

所以, 上面三个等式能成立, 只有 $x_1 = x_3 = 0$ 而 x_2 是任意非 0 值。不妨设 x_2 等于 1, 这样 \mathbf{u}_2 也具有单位长度。因此, $\mathbf{u}_2^T = (0 \ 1 \ 0)$ 。

38

由于 g 是对称的, 所以 $gg^T = g^Tg$, 则 gg^T 的特征向量与 g^Tg 的特征向量相同。那么 g 的 SVD 为:

$$\begin{aligned} g &= \sqrt{\lambda_1} \mathbf{u}_1 \mathbf{u}_1^T + \sqrt{\lambda_2} \mathbf{u}_2 \mathbf{u}_2^T = 2 \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{aligned}$$

这两个矩阵即为 g 的特征图像。

例 2.13 对下面图像进行奇异值分解, 并确定它的特征图像:

该矩阵的特征值由下得出

$$\begin{aligned} g &= \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \\ gg^T &= \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \end{pmatrix} \\ \begin{vmatrix} 1-\lambda & 0 & 1 \\ 0 & 2-\lambda & 0 \\ 1 & 0 & 1-\lambda \end{vmatrix} &= 0 \Rightarrow (1-\lambda)^2(2-\lambda) - (2-\lambda) = 0 \\ \Rightarrow (2-\lambda) \left[(1-\lambda)^2 - 1 \right] &= 0 \Rightarrow (2-\lambda)(1-\lambda-1)(1-\lambda+1) = 0 \end{aligned}$$

其中 $\lambda_1 = 2, \lambda_2 = 2, \lambda_3 = 0$ 。

第一个特征向量为

$$\begin{vmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \end{vmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = 2 \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \Rightarrow \begin{array}{l} x_1 + x_3 = 2x_1 \\ 2x_2 = 2x_2 \\ x_1 + x_3 = 2x_3 \end{array} \left| \begin{array}{l} x_1 = x_3 \\ x_2 \text{ 为任意值} \end{array} \right.$$

39

令 $x_1 = x_3 = \frac{1}{\sqrt{2}}$, $x_2 = 0$, 所以 $\mathbf{u}_1 = (\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}})^T$ 。

第二个特征向量必须满足相同的约束且必须与 \mathbf{u}_1 正交。因此

$$\mathbf{u}_2 = (0, 1, 0)^T$$

计算 $g^T g$ 相应的特征向量, 应用 (见例2.7):

$$\mathbf{v}_1 = g^T \mathbf{u}_1$$

因此

$$\mathbf{v}_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \sqrt{2} \\ 0 \end{pmatrix}$$

将其标准化以使 $|\mathbf{v}_1| = 1$; 也就是令 $\mathbf{v}_1 = (0, 1, 0)^T$ 。

$$\mathbf{v}_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

将其标准化以使 $|\mathbf{v}_2| = 1$; 也就是令 $\mathbf{v}_2 = (\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}})^T$ 。那么 g 的SVD就是

$$\begin{aligned} g &= \sqrt{\lambda_1} \mathbf{u}_1 \mathbf{v}_1^T + \sqrt{\lambda_2} \mathbf{u}_2 \mathbf{v}_2^T \\ &= \sqrt{2} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} (0 \ 1 \ 0) + \sqrt{2} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} (\frac{1}{\sqrt{2}} \ 0 \ \frac{1}{\sqrt{2}}) \\ &= \sqrt{2} \begin{pmatrix} 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 \end{pmatrix} + \sqrt{2} \begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \end{aligned}$$

40

这两个矩阵即为 g 的特征图像。

例2.14 验证下面图像奇异值分解的过程:

$$g = \begin{pmatrix} 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 100 & 100 & 100 & 255 & 255 \\ 255 & 255 & 100 & 150 & 150 & 150 & 100 & 255 \\ 255 & 255 & 100 & 150 & 200 & 150 & 100 & 255 \\ 255 & 255 & 100 & 150 & 150 & 150 & 100 & 255 \\ 255 & 255 & 255 & 100 & 100 & 100 & 255 & 255 \\ 255 & 255 & 255 & 255 & 50 & 255 & 255 & 255 \\ 50 & 50 & 50 & 50 & 255 & 255 & 255 & 255 \end{pmatrix}$$

矩阵 gg^T 为

$$gg^T = \begin{pmatrix} 520200 & 401625 & 360825 & 373575 & 360825 & 401625 & 467925 & 311100 \\ 401625 & 355125 & 291075 & 296075 & 291075 & 355125 & 381125 & 224300 \\ 360825 & 291075 & 282575 & 290075 & 282575 & 291075 & 330075 & 205025 \\ 373575 & 296075 & 290075 & 300075 & 290075 & 296075 & 332575 & 217775 \\ 360825 & 291075 & 282575 & 290075 & 282575 & 291075 & 330075 & 205025 \\ 401625 & 355125 & 291075 & 296075 & 291075 & 355125 & 381125 & 224300 \\ 467925 & 381125 & 330075 & 332575 & 330075 & 381125 & 457675 & 258825 \\ 311100 & 224300 & 205025 & 217775 & 205025 & 224300 & 258825 & 270100 \end{pmatrix}$$

它的特征值按降序排列为

$$\begin{matrix} 2593416.500 & 111621.508 & 71738.313 & 34790.875 \\ 11882.712 & 0.009 & 0.001 & 0.000 \end{matrix}$$

最后3个特征值几乎为0, 所以我们只计算前5个特征值对应的特征向量。这些特征向量就是下面矩阵的列:

$$\begin{pmatrix} -0.441 & 0.167 & 0.080 & -0.388 & 0.764 \\ -0.359 & -0.252 & 0.328 & 0.446 & 0.040 \\ -0.321 & -0.086 & -0.440 & 0.034 & -0.201 \\ -0.329 & -0.003 & -0.503 & 0.093 & 0.107 \\ -0.321 & -0.086 & -0.440 & 0.035 & -0.202 \\ -0.359 & -0.252 & 0.328 & 0.446 & 0.040 \\ -0.407 & -0.173 & 0.341 & -0.630 & -0.504 \\ -0.261 & 0.895 & 0.150 & 0.209 & -0.256 \end{pmatrix}$$

41

矩阵 $g^T g$ 为

$$g^T g = \begin{pmatrix} 457675 & 457675 & 339100 & 298300 & 269025 & 308550 & 349350 & 467925 \\ 457675 & 457675 & 339100 & 298300 & 269025 & 308550 & 349350 & 467925 \\ 339100 & 339100 & 292600 & 228550 & 191525 & 238800 & 302850 & 349350 \\ 298300 & 298300 & 228550 & 220050 & 185525 & 230300 & 238800 & 308550 \\ 269025 & 269025 & 191525 & 185525 & 237550 & 237800 & 243800 & 321300 \\ 308550 & 308550 & 238800 & 230300 & 237800 & 282575 & 291075 & 360825 \\ 349350 & 349350 & 302850 & 238800 & 243800 & 291075 & 355125 & 401625 \\ 467925 & 467925 & 349350 & 308550 & 321300 & 360825 & 401625 & 520200 \end{pmatrix}$$

它的特征向量经单独计算, 结果为下列矩阵的列:

$$\begin{pmatrix} -0.410 & -0.389 & 0.264 & 0.106 & 0.012 \\ -0.410 & -0.389 & 0.264 & 0.106 & 0.012 \\ -0.316 & -0.308 & -0.537 & -0.029 & -0.408 \\ -0.277 & -0.100 & 0.101 & -0.727 & -0.158 \\ -0.269 & 0.555 & 0.341 & 0.220 & -0.675 \\ -0.311 & 0.449 & -0.014 & -0.497 & 0.323 \\ -0.349 & 0.241 & -0.651 & 0.200 & 0.074 \\ -0.443 & 0.160 & 0.149 & 0.336 & 0.493 \end{pmatrix}$$

图2-1展示了原始图像和它的5幅特征图像。每一幅特征图像的灰度值已被标定以使它可以在0到255之间变化。这些特征图像需由相应的特征值的平方根加权再相加, 生成原始图像。

图2-2中所示的5幅图像是当1、2、3、4、5个特征值用于重构时得到的重构图像。

计算每个重构图像平方误差的和可根据下面公式：

$$\sum_{i=1}^{64} (\text{重构像素} - \text{原始像素})^2$$

可得：

图像a) 的误差：230033.32 ($\lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 = 230033.41$)

图像b) 的误差：118412.02 ($\lambda_3 + \lambda_4 + \lambda_5 = 118411.90$)

图像c) 的误差：46673.53 ($\lambda_4 + \lambda_5 = 46673.59$)

图像d) 的误差：11882.65 ($\lambda_5 = 11882.71$)

图像e) 的误差：0

可以看出忽略掉的特征值的和恰好与重构图像的误差相一致。

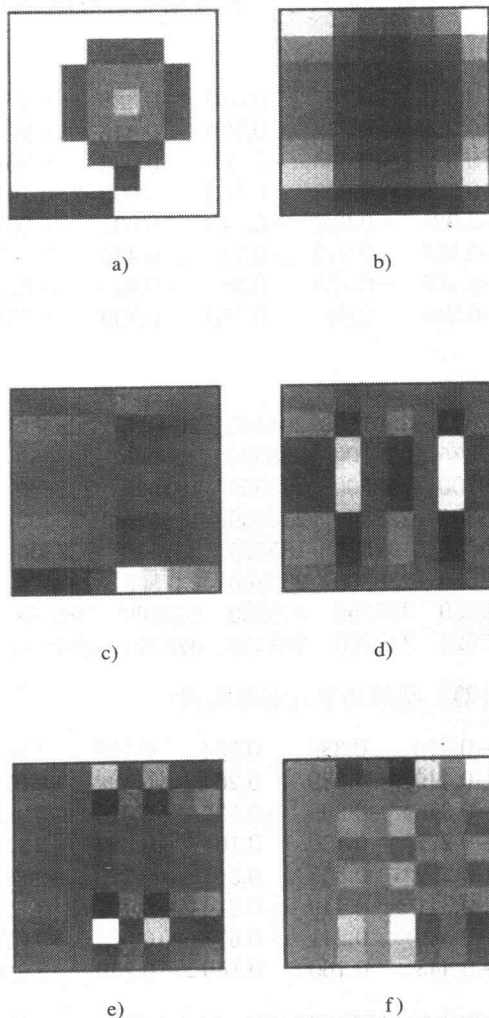


图2-1 原始图像与其5个特征图像，每一个标定值从0到255

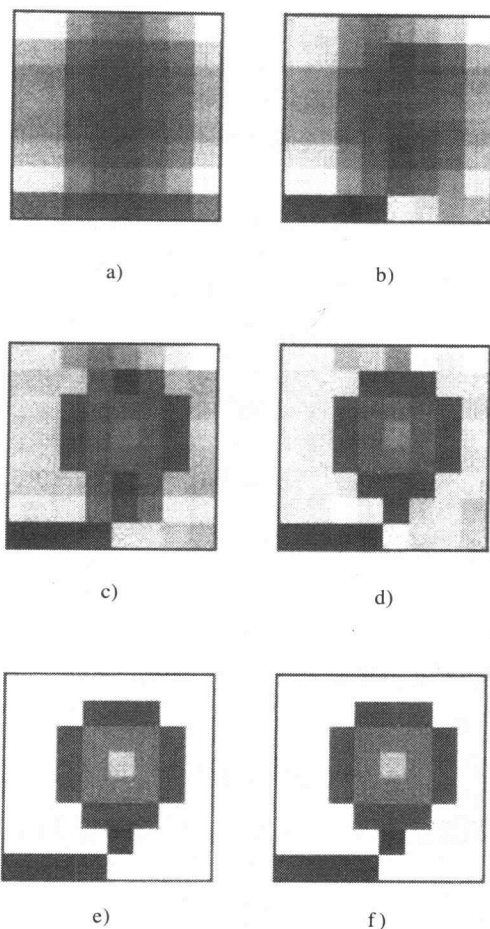


图2-2 从右上到左下依次用1、2、3、4、5个特征值
重构时得到的重构图像，图2-2 f是原始图像

2.17 任意被展开的图像是否存在基图像集

是。它们可以由离散化的完全标准正交函数集确定。

2.18 什么是函数的完全标准正交集

函数集 $S_n(t)$ ，其中 n 是整数，被称为是在区间 $[0, T]$ 上带权函数 $w(t)$ 正交的 (orthogonal) 函数集，如果：

$$\int_0^T w(t) S_n(t) S_m(t) dt = \begin{cases} k & \text{如果 } n = m \\ 0 & \text{如果 } n \neq m \end{cases} \quad (2-31)$$

如果 $k = 1$ ，这个集称为标准正交的 (orthonormal)。如果不能找到任何其他函数与该集正交且不属于该集，它被称为完全的 (complete)。一个完全标准正交集的例子就是傅里叶变换

的基函数集 e^{jnt} 。

• 例2.15 证明正交矩阵的列构成一个标准正交向量集。

• 不妨设 A 是一个 $N \times N$ 的正交矩阵 (也就是 $A^T = A^{-1}$)，设它的列向量是 $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N$ 。显然有：

$$A^{-1}A = I \Rightarrow A^T A = I \Rightarrow \begin{pmatrix} \mathbf{u}_1^T \\ \mathbf{u}_2^T \\ \vdots \\ \mathbf{u}_N^T \end{pmatrix} (\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_N) = I \Rightarrow$$

$$\begin{pmatrix} \mathbf{u}_1^T \mathbf{u}_1 & \mathbf{u}_1^T \mathbf{u}_2 & \dots & \mathbf{u}_1^T \mathbf{u}_N \\ \mathbf{u}_2^T \mathbf{u}_1 & \mathbf{u}_2^T \mathbf{u}_2 & \dots & \mathbf{u}_2^T \mathbf{u}_N \\ \vdots & \vdots & & \vdots \\ \mathbf{u}_N^T \mathbf{u}_1 & \mathbf{u}_N^T \mathbf{u}_2 & \dots & \mathbf{u}_N^T \mathbf{u}_N \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

45 这证明了 A 的列构成一个标准正交向量集。

例2.16 证明正交矩阵的逆也是正交的。

正交矩阵定义如下：

$$A^T = A^{-1} \quad (2-32)$$

欲证明 A^{-1} 是正交的，只需证明 $(A^{-1})^T = (A^{-1})^{-1}$ 。这等价于 $(A^{-1})^T = A$ ，而这很容易由对式(2-32)取转置导出。

例2.17 证明正交矩阵的行也构成一个标准正交向量集。

因为 A 是正交矩阵，所以 A^{-1} 也是 (见例2.16)。一个正交矩阵的列构成一个标准正交向量集 (见例2.15)。因此， A^{-1} 的列，也就是 A 的行，构成一个标准正交向量集。

2.19 是否存在标准正交离散函数完全集

是的。存在，例如，取值于集合 $\{0, \pm 1, \pm \sqrt{2}^p, p=1, 2, 3, \dots\}$ 的哈尔函数集(Haar function)和取值于集合 $\{+1, -1\}$ 的沃尔什函数 (Walsh function)。

2.20 如何定义哈尔函数

$$H_0(t) = 1 \quad \text{当 } 0 \leq t < 1 \text{ 时}$$

$$H_1(t) = \begin{cases} 1 & \text{如果 } 0 \leq t < \frac{1}{2} \\ -1 & \text{如果 } \frac{1}{2} \leq t < 1 \end{cases}$$

$$H_{2^{p+n}}(t) = \begin{cases} \sqrt{2}^p & \text{当 } \frac{n}{2^p} \leq t < \frac{(n+0.5)}{2^p} \text{ 时} \\ -\sqrt{2}^p & \text{当 } \frac{(n+0.5)}{2^p} \leq t < \frac{(n+1)}{2^p} \text{ 时} \\ 0 & \text{其他} \end{cases} \quad (2-33) \quad \boxed{46}$$

其中 $p = 1, 2, 3, \dots, n = 0, 1, \dots, 2^p - 1$ 。

2.21 如何定义沃尔什函数

可以通过各种形式定义，且可以证明是等价的。现在利用差分方程来定义：

$$W_{2j+q}(t) = (-1)^{\lfloor \frac{j}{2} \rfloor + q} \{W_j(2t) + (-1)^{j+q} W_j(2t-1)\} \quad (2-34)$$

其中 $\lfloor \frac{j}{2} \rfloor$ 是取整函数，表示不超过 $\frac{j}{2}$ 的最大整数， $q = 0$ 或 1 ， $j = 0, 1, 2, \dots$

$$W_0(t) = \begin{cases} 1 & \text{当 } 0 \leq t < 1 \text{ 时} \\ 0 & \text{其他} \end{cases}$$

上述方程按所谓的自然序数 (natural order) 定义这些函数。其他等价定义是按照函数零交叉点数递增排列，也就是所谓的列率序 (sequency order)。

2.22 如何从哈尔函数和沃尔什函数创建一个图像变换矩阵

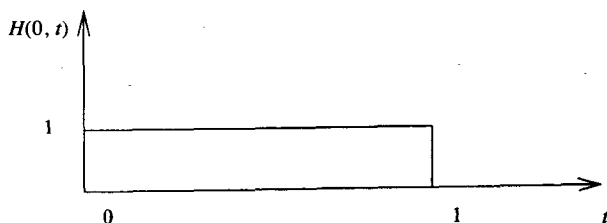
首先按照所要创建矩阵的大小来标定独立变量 t ，然后仅考虑它的整数值 i 。对于 $k = 0, 1, 2, \dots, N-1, i = 0, 1, \dots, N-1$ ， $H_k(i)$ 可以写成矩阵形式，且可用于二维离散图像函数的变换。对 $W_k(i)$ 可类似进行。

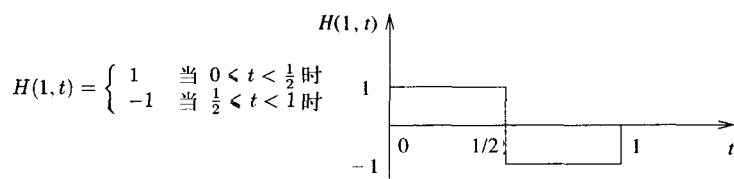
注意这种方式定义的哈尔函数/沃尔什函数不是正交的，每一个都要经过标准化，连续情况乘以 $\frac{1}{\sqrt{T}}$ ；离散情况乘以 $\frac{1}{\sqrt{N}}$ ，其中 t 取 N 等分的离散值。

• 例2.18 推导出可以用于计算一个 4×4 图像的哈尔变换的矩阵。

• 首先，利用式(2-33)，可以计算并画出具有连续变量 t 的哈尔函数，这是计算变换矩阵所必须的。

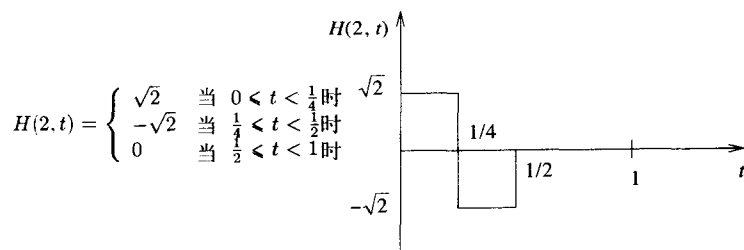
$$H(0, t) = 1 \quad \text{当 } 0 \leq t < 1 \text{ 时}$$



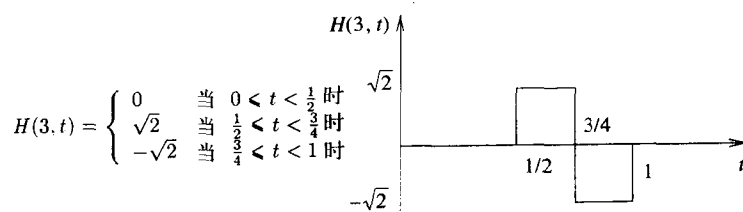


在哈尔函数定义中, 当 $p = 1$ 时, n 取值 0 和 1:

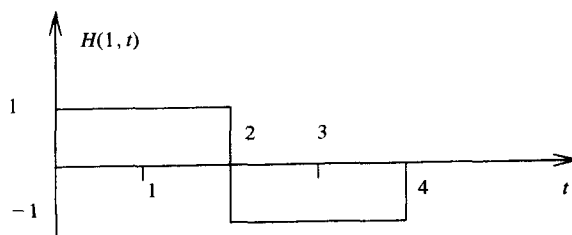
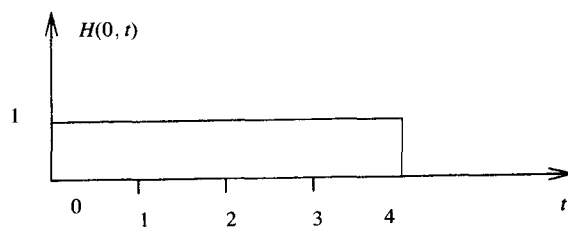
如果 $p = 1, n = 0$:

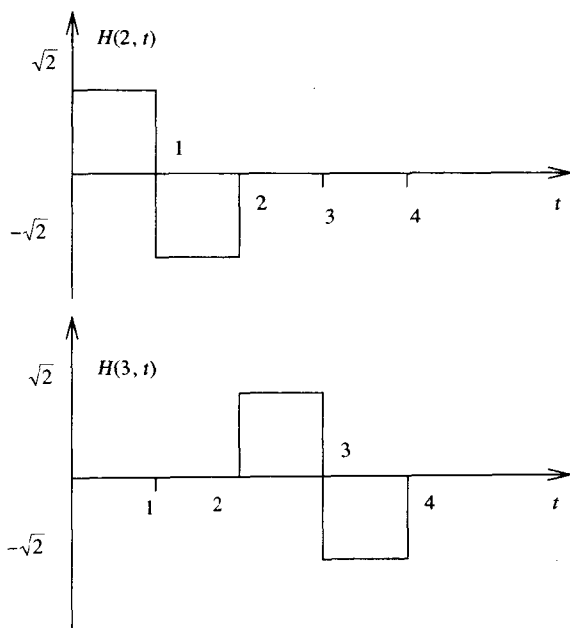


如果 $p = 1, n = 1$:



对一个 4×4 的图像做变换需要一个 4×4 的矩阵。如果将 t 轴乘以 4 改变它的比例, 并且只取 t 的整数值 (也就是 $t = 0, 1, 2, 3$), 就能够建立变换矩阵。伸缩后的函数如图所示。





变换矩阵中的元素为函数 $H(s, t)$ 的值, 其中 s 和 t 取值为 0、1、2、3。变换矩阵为

$$H = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{pmatrix}$$

系数 $\frac{1}{2}$ 使 $HH^T = I$, 即单位矩阵。

49

例2.19 计算下面图像的哈尔变换:

$$g = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

图像 g 的哈尔变换为 $A = HgH^T$ 。下面利用在例2.18中导出的矩阵 H :

$$\begin{aligned} A &= \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & \sqrt{2} & 0 \\ 1 & 1 & -\sqrt{2} & 0 \\ 1 & -1 & 0 & \sqrt{2} \\ 1 & -1 & 0 & -\sqrt{2} \end{pmatrix} \\ &= \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{pmatrix} \begin{pmatrix} 2 & 0 & -\sqrt{2} & \sqrt{2} \\ 2 & 0 & \sqrt{2} & -\sqrt{2} \\ 2 & 0 & \sqrt{2} & -\sqrt{2} \\ 2 & 0 & -\sqrt{2} & \sqrt{2} \end{pmatrix} \end{aligned}$$

$$\begin{aligned}
 &= \frac{1}{4} \begin{pmatrix} 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -4 & 4 \\ 0 & 0 & 4 & -4 \end{pmatrix} \\
 &= \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}
 \end{aligned}$$

例2.20 通过将交换矩阵的右下角元素置为0, 应用近似哈尔变换, 重构例2.19中的图像。

50 近似变换矩阵变为:

$$\tilde{A} = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

重构图像由 $\tilde{g} = H^T \tilde{A} H$ 得出:

$$\begin{aligned}
 \tilde{g} &= \frac{1}{4} \begin{pmatrix} 1 & 1 & \sqrt{2} & 0 \\ 1 & 1 & -\sqrt{2} & 0 \\ 1 & -1 & 0 & \sqrt{2} \\ 1 & -1 & 0 & -\sqrt{2} \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{pmatrix} \\
 &= \frac{1}{4} \begin{pmatrix} 1 & 1 & \sqrt{2} & 0 \\ 1 & 1 & -\sqrt{2} & 0 \\ 1 & -1 & 0 & \sqrt{2} \\ 1 & -1 & 0 & -\sqrt{2} \end{pmatrix} \begin{pmatrix} 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 \\ -\sqrt{2} & \sqrt{2} & \sqrt{2} & -\sqrt{2} \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \end{pmatrix} \\
 &= \frac{1}{4} \begin{pmatrix} 0 & 4 & 4 & 0 \\ 4 & 0 & 0 & 4 \\ 4 & 0 & 2 & 2 \\ 0 & 4 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0.5 & 0.5 \\ 0 & 1 & 0 & 0 \end{pmatrix}
 \end{aligned}$$

平方误差等于:

$$0.5^2 + 0.5^2 + 1^2 = 1.5$$

•
•
•

2.23 哈尔变换的基图像是怎样的

图2-3表示一个 8×8 的图像按哈尔函数展开的基图像。这些图像都是由离散哈尔函数自身或与另外一个取外积所产生的。沿左端和下端的数字表示分别沿每一行或列所用函数的顺序。

51 为了便于展示, 每个图像的离散值已被规定在 $[0, 255]$ 范围内变化。

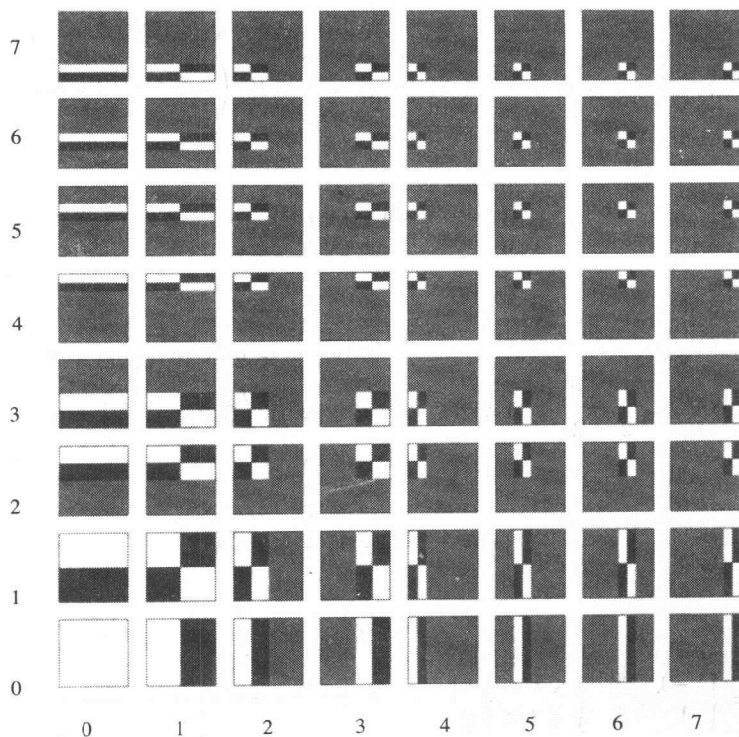
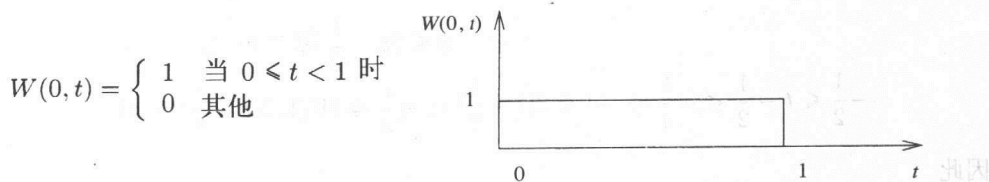


图2-3 哈尔变换的基图像

在每一幅图像中，灰色代表0，黑色代表一个负数，白色代表一个正数。注意每幅图像已分别标定：从一幅图像到下一幅图像黑色和白色代表不同的值。

• 例2.21 推导一个可用于沃尔什变换的 4×4 图像。

首先，应用式(2-34)计算并绘制连续变量 t 的沃尔什函数，这在转换矩阵的计算中是必须的。



$j = 0, q = 1, \left[\frac{j}{2}\right] = 0$ 的情况:

$$W(1, t) = - \left\{ W(0, 2t) - W\left(0, 2\left(t - \frac{1}{2}\right)\right) \right\}$$

当 $0 \leq t < \frac{1}{2}$ 时:

$$0 \leq 2t < 1 \Rightarrow W(0, 2t) = 1$$

$$-\frac{1}{2} \leq t - \frac{1}{2} < 0 \Rightarrow -1 \leq 2(t - \frac{1}{2}) < 0 \Rightarrow W\left(0, 2(t - \frac{1}{2})\right) = 0$$

因此

$$W(1, t) = -1, \quad 0 \leq t < \frac{1}{2}$$

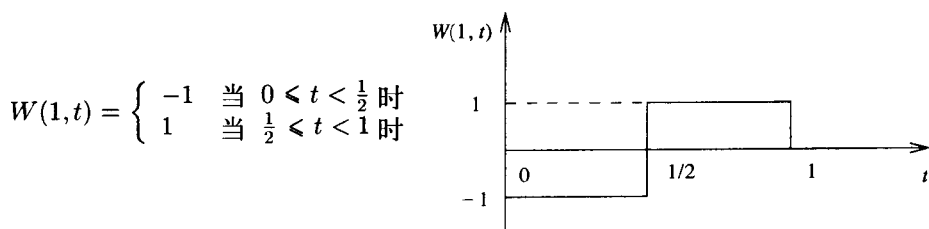
当 $\frac{1}{2} \leq t < 1$ 时:

$$1 \leq 2t < 2 \Rightarrow W(0, 2t) = 0$$

$$0 \leq t - \frac{1}{2} < \frac{1}{2} \Rightarrow 0 \leq 2(t - \frac{1}{2}) < 1 \Rightarrow W\left(0, 2(t - \frac{1}{2})\right) = 1$$

因此

$$W(1, t) = -(-1) = 1, \quad \frac{1}{2} \leq t < 1$$



$j=1, q=0, \left[\frac{j}{2}\right] = 0$ 的情况:

$$W(2, t) = W(1, 2t) - W\left(1, 2(t - \frac{1}{2})\right)$$

53

当 $0 \leq t < \frac{1}{4}$ 时:

$$0 \leq 2t < \frac{1}{2} \Rightarrow W(1, 2t) = -1$$

$$-\frac{1}{2} \leq t - \frac{1}{2} < -\frac{1}{4} \Rightarrow -1 \leq 2(t - \frac{1}{2}) < -\frac{1}{2} \Rightarrow W\left(1, 2(t - \frac{1}{2})\right) = 0$$

因此

$$W(2, t) = -1, \quad 0 \leq t < \frac{1}{4}$$

当 $\frac{1}{4} \leq t < \frac{1}{2}$ 时:

$$\frac{1}{2} \leq 2t < 1 \Rightarrow W(1, 2t) = 1$$

$$-\frac{1}{4} \leq t - \frac{1}{2} < 0 \Rightarrow -\frac{1}{2} \leq 2(t - \frac{1}{2}) < 0 \Rightarrow W\left(1, 2(t - \frac{1}{2})\right) = 0$$

因此

$$W(2, t) = 1, \quad \frac{1}{4} \leq t < \frac{1}{2}$$

当 $\frac{1}{2} \leq t < \frac{3}{4}$ 时:

$$1 \leq 2t < \frac{3}{2} \Rightarrow W(1, 2t) = 0$$

$$0 \leq t - \frac{1}{2} < \frac{1}{4} \Rightarrow 0 \leq 2(t - \frac{1}{2}) < \frac{1}{2} \Rightarrow W\left(1, 2(t - \frac{1}{2})\right) = -1$$

因此

$$W(2, t) = 1, \quad \frac{1}{2} \leq t < \frac{3}{4}$$

当 $\frac{3}{4} \leq t < 1$ 时:

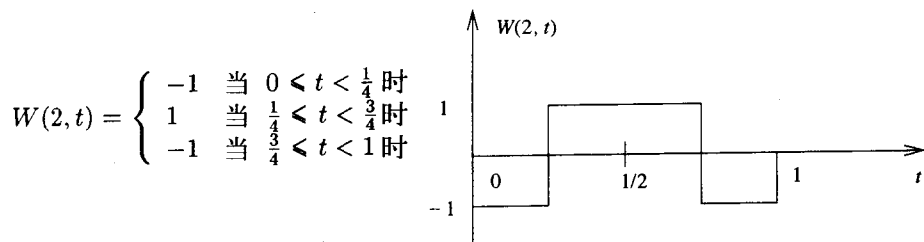
$$\frac{3}{2} \leq 2t < 2 \Rightarrow W(1, 2t) = 0$$

$$\frac{1}{4} \leq t - \frac{1}{2} < \frac{1}{2} \Rightarrow \frac{1}{2} \leq 2(t - \frac{1}{2}) < 1 \Rightarrow W\left(1, 2(t - \frac{1}{2})\right) = 1$$

因此

$$W(2, t) = -1, \quad \frac{3}{4} \leq t < 1$$

54



$j=1, q=1 \left\lceil \frac{j}{2} \right\rceil = 0$ 的情况:

$$W(3, t) = - \left\{ W(1, 2t) + W\left(1, 2(t - \frac{1}{2})\right) \right\}$$

当 $0 \leq t < \frac{1}{4}$ 时:

$$W(1, 2t) = -1, \quad W\left(1, 2(t - \frac{1}{2})\right) = 0$$

因此

$$W(3, t) = 1, \quad 0 \leq t < \frac{1}{4}$$

当 $\frac{1}{4} \leq t < \frac{1}{2}$ 时:

$$W(1, 2t) = 1, \quad W\left(1, 2\left(t - \frac{1}{2}\right)\right) = 0$$

因此

$$W(3, t) = -1, \quad \frac{1}{4} \leq t < \frac{1}{2}$$

当 $\frac{1}{2} \leq t < \frac{3}{4}$ 时:

$$W(1, 2t) = 0, \quad W\left(1, 2\left(t - \frac{1}{2}\right)\right) = -1$$

因此

$$W(3, t) = 1, \quad \frac{1}{2} \leq t < \frac{3}{4}$$

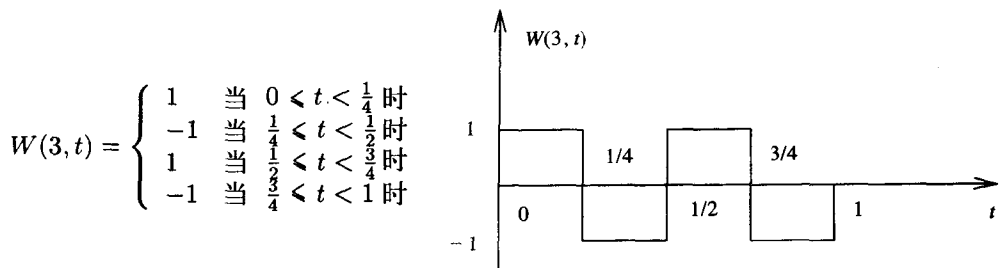
当 $\frac{3}{4} \leq t < 1$ 时:

$$W(1, 2t) = 0, \quad W\left(1, 2\left(t - \frac{1}{2}\right)\right) = 1$$

55

因此

$$W(3, t) = -1, \quad \frac{3}{4} \leq t < 1$$



为了得到一个 4×4 的矩阵, 将 t 乘以 4 并且只考虑 t 的整数值, 也就是 0、1、2、3。
 $W(0, t)$ 的值作为矩阵的第一行, $W(1, t)$ 作为矩阵的第二行, $W(2, t)$ 为第三行, 依此类推:

$$W = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 \end{pmatrix}$$

令该矩阵乘以 $\frac{1}{2}$ 得到其标准化形式。因此 $W^T W = I$, 其中, I 是单位矩阵。

例2.22 计算以下矩阵的沃尔什变换:

$$g = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

56

在一般的可分离的线性变换公式 $A = UgV^T$ 中, 令 $U = V = W$, 其中 W 是例2.21中推导的矩阵:

$$\begin{aligned} A &= \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \end{pmatrix} \\ &= \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 2 & 0 \\ 2 & 0 & -2 & 0 \\ 2 & 0 & -2 & 0 \\ 2 & 0 & 2 & 0 \end{pmatrix} \\ &= \frac{1}{4} \begin{pmatrix} 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -8 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{aligned}$$

...

2.24 如何仅用+1或-1定义一个正交矩阵

我们可以做到这一点。哈达玛矩阵 (Hadamard matrix) 由数学家哈达玛于1893年提出而得名。哈达玛矩阵由其维数以递归的形式来定义, 而该维数是2的幂, 如下所示:

$$H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad H_{2N} = \begin{pmatrix} H_N & H_N \\ H_N & -H_N \end{pmatrix}$$

在这样的定义下, 正交矩阵的行可看作由离散形式的沃尔什函数值经过一定顺序排列得到。因此当 $N = 2^n$ 时, 沃尔什函数可以根据这些矩阵来计算。这种方式产生的沃尔什函数称为克罗内克 (Kronecker) 序或字典序 (lexicographic), 也就是说这时的排列次序与利用式(2-34)产生的排列次序不同。

2.25 哈达玛/沃尔什变换的基图像是怎样的

57

图2-4是 8×8 大小、按克罗内克序排列的沃尔什函数展开的二值基图像。

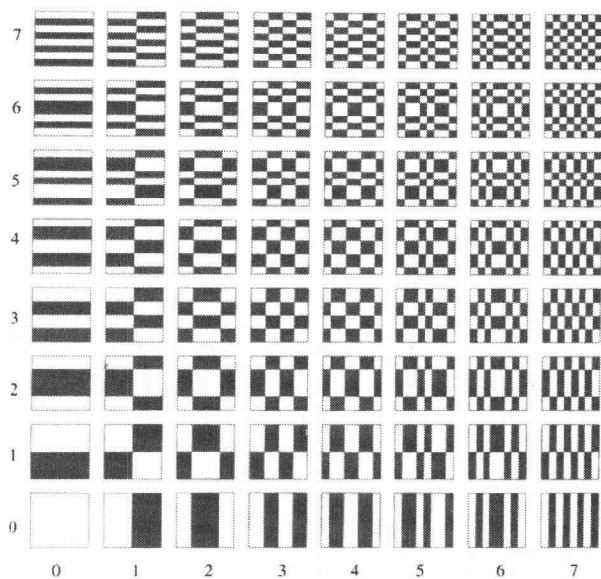


图2-4 哈达玛/沃尔什变换的基图像

● 例2.23 写出例2.14中图像的哈尔变换的过程:

图2-5所示的8个图像均是重构后的图像，它们分别利用了1、2、……、8个哈尔函数产生的基图像进行重构。例如，仅保留了图2-3中左下角的4个基图像，再乘以相应的系数便得到图2-5b的重构后的图像。这4个基图像由前两个哈尔函数 $H(0, t)$ 和 $H(1, t)$ 产生。把图2-3中第一行和第一列的基图像各自所对应的系数全置为0，便重构图2-5g所示的图像。所使用的基图像由前7个哈尔函数产生，也就是 $H(0, t)$ 、 $H(1, t)$ 、……、 $H(6, t)$ 。

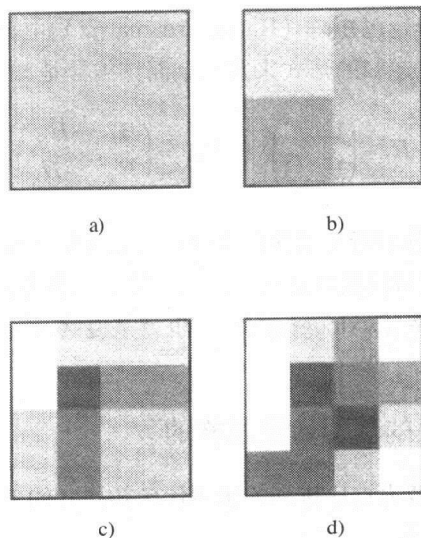


图2-5 重构后的图像

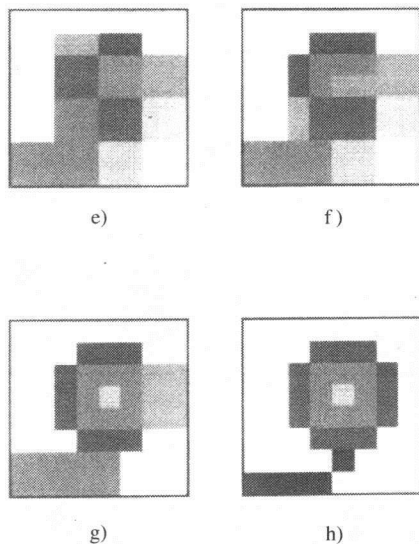


图2-5 (续)

所用的基图像从左上到右下, 分别由前1、2、3、……、8个哈尔函数产生。

59

重构图像的平方误差和分别为:

图2-5a的误差: 366394

图2-5b的误差: 356192

图2-5c的误差: 291740

图2-5d的误差: 222550

图2-5e的误差: 192518

图2-5f的误差: 174625

图2-5g的误差: 141100

图2-5h的误差: 0

例2.24 写出例2.14中图像的沃尔什/哈达玛变换的过程:

图2-6所示的8个图像均是重构图像, 分别使用了1、2、……、8个沃尔什函数所产生的基图像来重构。例如, 图2-6f利用了沃尔什/哈达玛逆变换来重构, 其过程是把图2-4中的第1、2行以及最右边两列的基图像相乘所得的变换矩阵的元素置为0。被忽略了的基图像正是由函数 $W(6, t)$ 以及 $W(7, t)$ 产生的。

重构图像平方误差和分别为:

图2-6a的误差: 366394

图2-6b的误差: 356190

图2-6c的误差: 262206

图2-6d的误差: 222550

60

图2-6e的误差: 148029

图2-6f的误差: 92078

图2-6g的误差: 55905

图2-6h的误差: 0

60

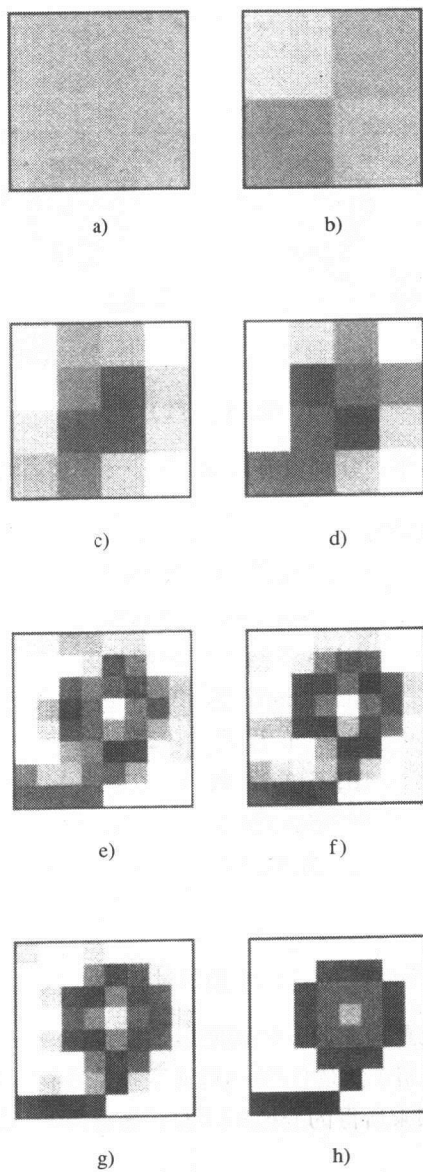


图2-6 重构后的图像

61

使用的基图像从左上到右下, 分别由前1、2、3、……、8个沃尔什函数产生。

•
•
•

2.26 沃尔什和哈尔变换的优缺点是什么

从图2-3可以看出,哈尔变换的基图像以相同的基本方式扫描整幅待变换的图像,每扫描一次仅获取原图像的一个局部,基图像的序号越大,获得的特征越精确。例如,图2-3右上方中,占全图1/4的16个基图像,均使用 2×2 大小的窗口来复制原图像中不同位置的细节。假如我们对这种尺度上的细节不感兴趣,可以将哈尔变换中16个相应的系数置为零。因此,截断一个哈尔重构时,必须保留前 2^0 、 2^1 、 2^2 、……个哈尔函数产生的基图像,而不是任意多个。用这种方法,重构后的误差才会完全一致地分配到整个图像中。作为另一种选择,我们希望保留能对这些位置的重构精度起作用的基图像(也就是保留基图像对应的系数)。换句话说,哈尔基函数能以不同精度来重构原图不同位置的细节。

相反,沃尔什基图像的序号越高,整体上就越接近原图像,并且不同位置的细节精度完全一致。其原因是沃尔什函数不能取零值。注意,两种基函数之间的差别正好反映在重构图像上:图2-5g与图2-6g都保留了相同个数的基图像。图2-5g中的花,除了右方和下方的部分细节外,几乎完全被重构。因为被省略的基图像正是捕捉原图像此位置细节的,而原图在此处恰好具有明显的细节结构。这就是重构的误差在使用哈尔函数时比使用沃尔什函数时要大的原因。注意,沃尔什函数的重构误差被完全一致地分配到整个图像中。

另一方面,沃尔什变换比哈尔变换优势在于,沃尔什变换只取+1和-1两个值,类似于二值的逻辑变量,因此易于在计算机上实现。

2.27 什么是哈尔小波

哈尔基函数能从部分中反映图像性质。这一属性是一类更一般的函数所特有的,它们被称为小波(wavelet)。哈尔小波(Haar wavelet)是对同一函数进行不同尺度伸缩和平移得到的各种形式。对于一个 8×8 图像来说,如图2-3所示,最上面7个以及最右边7个基图像是哈尔小波的图像,图2-3左下角的基图像实际上是平均图像,它所描述的函数被称为尺度函数(scaling function)。最后一行和第一列的其他基图像是由尺度函数和小波共同生成的。所有基图像构成了一组完备的基,在这种意义下,任何一个 8×8 图像均能展开。

62

2.28 傅里叶变换的离散形式是怎样的

在离散点 k ,函数 $f(k)$ 的一维离散傅里叶变换定义如下:

$$F(m) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} f(k) \exp \left[-j \frac{2\pi mk}{N} \right] \quad (2-35)$$

一个 $N \times N$ 图像的二维离散傅里叶变换(DFT)定义如下:

$$\alpha_{mn} = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} g_{kl} e^{-j2\pi \frac{km+nl}{N}} \quad (2-36)$$

与其他直接在离散域上导出的变换不同,傅里叶变换起初是在连续域上推导。为了保持这种“历史”的一致性,我们回到使用函数变量而不是标号的方式来定义。此外,考虑不同函数的傅里叶变换,规定函数在傅里叶变换前后使用相同字母表示,只需在函数名上方加一

小标记。小标记的个数用以区分同一函数的不同形式的傅里叶变换。当某些情况出现时就能清晰地知道我们这么做的原因。现在,定义一个 $M \times N$ 图像的傅里叶变换如下:

$$\hat{g}(m, n) = \frac{1}{\sqrt{MN}} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} g(k, l) e^{-2\pi j [\frac{km}{M} + \frac{ln}{N}]} \quad (2-37)$$

把此公式看作“自动贩卖机”(slot machine), 只要把函数“插”进式(2-37), 弹出结果便是它的DFT。

$$\underbrace{\quad}_{DFT} = \frac{1}{\sqrt{MN}} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} \underbrace{\quad}_{\text{函数}} e^{-2\pi j [\frac{km}{M} + \frac{ln}{N}]} \quad (2-38)$$

B2.3 离散傅里叶逆变换是怎样的?

在式(2-37)两边同时乘以 $e^{2\pi j [\frac{qm}{M} + \frac{pn}{N}]} \frac{1}{\sqrt{MN}}$, 并对所有的 m, n 分别从0到 $M-1$, 0到 $N-1$ 求和。得到

$$\begin{aligned} & \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \hat{g}(m, n) e^{2\pi j [\frac{qm}{M} + \frac{pn}{N}]} \\ &= \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} g(k, l) e^{2\pi j [\frac{m(q-k)}{M} + \frac{n(p-l)}{N}]} \\ &= \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} g(k, l) \sum_{m=0}^{M-1} e^{2\pi j m \frac{q-k}{M}} \sum_{n=0}^{N-1} e^{2\pi j n \frac{p-l}{N}} \end{aligned} \quad (2-39)$$

检验下面的和式:

$$\sum_{m=0}^{M-1} e^{2\pi j t \frac{m}{M}} \quad (2-40)$$

其中 t 是整数。这是一个 M 项的等比级数, 首项为1($m=0$), 公比 $q = e^{2\pi j \frac{t}{M}}$ 。该等比级数前 n 项和为:

$$\sum_{k=0}^{n-1} q^k = \frac{q^n - 1}{q - 1}, \quad q \neq 1 \quad (2-41)$$

因此, 式(2-40)的和等于:

$$\sum_{m=0}^{M-1} e^{2\pi j t \frac{m}{M}} = \frac{e^{2\pi j t} - 1}{e^{2\pi j \frac{t}{M}} - 1} = \frac{\cos 2\pi t + j \sin 2\pi t - 1}{e^{2\pi j \frac{t}{M}} - 1} = 0 \quad (2-42)$$

然而, 如果 $t=0$, 式(2-40)中的所有项相加等于1, 且 $\sum_{m=0}^{M-1} 1 = M$ 。所以:

$$\sum_{m=0}^{M-1} e^{2\pi j t \frac{m}{M}} = \begin{cases} M, & t=0 \\ 0, & t \neq 0 \end{cases} \quad (2-43)$$

首先令 $t \equiv q-k$, 将式(2-43)代入式(2-39)中, 然后再令 $t \equiv p-l$, 推导得到式(2-39)右边的形式:

$$\frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} g(k, l) M \delta(q-k) N \delta(p-l)$$

其中, 除非 $a=b$, 否则 $\delta(a-b)=0$ 。因此, 上面的表达式就是 $g(q, p)$, 也就是说:

$$g(q, p) = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \hat{g}(m, n) e^{2\pi j [\frac{qm}{M} + \frac{pn}{N}]} \quad (2-44)$$

这就是二维离散傅里叶逆变换。

64

2.29 离散傅里叶变换如何用矩阵形式表示

用下列元素构造矩阵 U

$$U_{x\alpha} = \frac{1}{\sqrt{N}} e^{-j \frac{2\pi x\alpha}{N}} \quad (2-45)$$

其中 x 按列分别取 $0, 1, \dots, N-1$, α 按行也分别取相同的值。注意用这种方法构造的矩阵是对称的; $U^T = U$ 。图像 g 的二维离散傅里叶变换形式如下:

$$A = U g U \quad (2-46)$$

• 例2.25 推导用于 4×4 图像的离散傅里叶逆变换的矩阵。

应用式(2-45), 其中 $N=4, 0 \leq x \leq 3, 0 \leq \alpha \leq 3$:

$$\begin{aligned} U &= \frac{1}{\sqrt{4}} \begin{pmatrix} e^{-j \frac{2\pi}{4} \times 0} & e^{-j \frac{2\pi}{4} \times 0} & e^{-j \frac{2\pi}{4} \times 0} & e^{-j \frac{2\pi}{4} \times 0} \\ e^{-j \frac{2\pi}{4} \times 0} & e^{-j \frac{2\pi}{4} \times 1} & e^{-j \frac{2\pi}{4} \times 2} & e^{-j \frac{2\pi}{4} \times 3} \\ e^{-j \frac{2\pi}{4} \times 0} & e^{-j \frac{2\pi}{4} \times 2} & e^{-j \frac{2\pi}{4} \times 0} & e^{-j \frac{2\pi}{4} \times 2} \\ e^{-j \frac{2\pi}{4} \times 0} & e^{-j \frac{2\pi}{4} \times 3} & e^{-j \frac{2\pi}{4} \times 2} & e^{-j \frac{2\pi}{4} \times 1} \end{pmatrix} \\ &= \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & e^{-\frac{\pi}{2}j} & e^{-j\pi} & e^{-\frac{3\pi}{2}j} \\ 1 & e^{-j\pi} & 1 & e^{-j\pi} \\ 1 & e^{-\frac{6\pi}{4}j} & e^{-j\pi} & e^{-\frac{2\pi}{4}j} \end{pmatrix} \end{aligned}$$

$$e^{-\frac{2\pi}{4}j} = e^{-\frac{\pi}{2}j} = \cos \frac{\pi}{2} - j \sin \frac{\pi}{2} = -j$$

$$e^{-\pi j} = \cos \pi - j \sin \pi = -1$$

$$e^{-\frac{6\pi}{4}j} = \cos \frac{3\pi}{2} - j \sin \frac{3\pi}{2} = j$$

所以

$$U = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix}$$

65

例2.26 利用例2.25中矩阵 U ，计算以下图像的离散傅里叶变换：

$$g = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

首先计算 gU ：

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 \end{pmatrix}$$

对上面的结果从左乘 U ，得到 $UgU = A$ (图像的离散傅里叶变换)：

$$\begin{aligned} & \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix} \begin{pmatrix} 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 \end{pmatrix} \\ &= \frac{1}{4} \begin{pmatrix} 4 & -4 & 4 & -4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{aligned}$$

•
•
•

2.30 矩阵 U 可否用于DFT的酉矩阵

必须证明矩阵 U 的任意行与其他任意行正交。因为 U 为一复矩阵，检验正交性时，相乘的两行中的一行必须用其复共轭形式。 $x = x_1, x = x_2$ 时，两行的乘积如下：

$$\sum_{\alpha=0}^{N-1} e^{-j\frac{2\pi x_1 \alpha}{N}} e^{j\frac{2\pi x_2 \alpha}{N}} = \sum_{\alpha=0}^{N-1} e^{-j\frac{2\pi(x_1 - x_2)\alpha}{N}} \quad (2-47)$$

可以看出，上述和式正好是首项为1，公比为 $e^{-j\frac{2\pi(x_1 - x_2)\alpha}{N}}$ 的 N 项等比级数之和。可以证明此和为零。

66

如果 U 是酉矩阵, 则 U 每行的模为1。式(2-47)中, 如果令 $x_2 = x_1$, 所有行模的平方和为:

$$\sum_{\alpha=0}^{N-1} e^{-j\frac{2\pi x_1 \alpha}{N}} e^{j\frac{2\pi x_1 \alpha}{N}} = \sum_{\alpha=0}^{N-1} 1 = N \quad (2-48)$$

这样就消去了在标准化时矩阵所乘的常数项。

例2.27 推导用于计算 8×8 图像DFT的矩阵 U :

应用式(2-45), $N = 8$ 时有:

$$\frac{1}{\sqrt{8}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & e^{-j\frac{2\pi}{8} \times 1} & e^{-j\frac{2\pi}{8} \times 2} & e^{-j\frac{2\pi}{8} \times 3} & e^{-j\frac{2\pi}{8} \times 4} & e^{-j\frac{2\pi}{8} \times 5} & e^{-j\frac{2\pi}{8} \times 6} & e^{-j\frac{2\pi}{8} \times 7} \\ 1 & e^{-j\frac{2\pi}{8} \times 2} & e^{-j\frac{2\pi}{8} \times 4} & e^{-j\frac{2\pi}{8} \times 6} & e^{-j\frac{2\pi}{8} \times 8} & e^{-j\frac{2\pi}{8} \times 10} & e^{-j\frac{2\pi}{8} \times 12} & e^{-j\frac{2\pi}{8} \times 14} \\ 1 & e^{-j\frac{2\pi}{8} \times 3} & e^{-j\frac{2\pi}{8} \times 6} & e^{-j\frac{2\pi}{8} \times 9} & e^{-j\frac{2\pi}{8} \times 12} & e^{-j\frac{2\pi}{8} \times 15} & e^{-j\frac{2\pi}{8} \times 18} & e^{-j\frac{2\pi}{8} \times 21} \\ 1 & e^{-j\frac{2\pi}{8} \times 4} & e^{-j\frac{2\pi}{8} \times 8} & e^{-j\frac{2\pi}{8} \times 12} & e^{-j\frac{2\pi}{8} \times 16} & e^{-j\frac{2\pi}{8} \times 20} & e^{-j\frac{2\pi}{8} \times 24} & e^{-j\frac{2\pi}{8} \times 28} \\ 1 & e^{-j\frac{2\pi}{8} \times 5} & e^{-j\frac{2\pi}{8} \times 10} & e^{-j\frac{2\pi}{8} \times 15} & e^{-j\frac{2\pi}{8} \times 20} & e^{-j\frac{2\pi}{8} \times 25} & e^{-j\frac{2\pi}{8} \times 30} & e^{-j\frac{2\pi}{8} \times 35} \\ 1 & e^{-j\frac{2\pi}{8} \times 6} & e^{-j\frac{2\pi}{8} \times 12} & e^{-j\frac{2\pi}{8} \times 18} & e^{-j\frac{2\pi}{8} \times 24} & e^{-j\frac{2\pi}{8} \times 30} & e^{-j\frac{2\pi}{8} \times 36} & e^{-j\frac{2\pi}{8} \times 42} \\ 1 & e^{-j\frac{2\pi}{8} \times 7} & e^{-j\frac{2\pi}{8} \times 14} & e^{-j\frac{2\pi}{8} \times 21} & e^{-j\frac{2\pi}{8} \times 28} & e^{-j\frac{2\pi}{8} \times 35} & e^{-j\frac{2\pi}{8} \times 42} & e^{-j\frac{2\pi}{8} \times 49} \end{pmatrix}$$

由于

$$e^{-j\frac{2\pi}{N} \times x} = e^{-j\frac{2\pi}{N} \times [\text{mod}_N(x)]}$$

上述矩阵可以简化成:

$$U = \frac{1}{\sqrt{8}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & e^{-j\frac{2\pi}{8} \times 1} & e^{-j\frac{2\pi}{8} \times 2} & e^{-j\frac{2\pi}{8} \times 3} & e^{-j\frac{2\pi}{8} \times 4} & e^{-j\frac{2\pi}{8} \times 5} & e^{-j\frac{2\pi}{8} \times 6} & e^{-j\frac{2\pi}{8} \times 7} \\ 1 & e^{-j\frac{2\pi}{8} \times 2} & e^{-j\frac{2\pi}{8} \times 4} & e^{-j\frac{2\pi}{8} \times 6} & e^{-j\frac{2\pi}{8} \times 0} & e^{-j\frac{2\pi}{8} \times 2} & e^{-j\frac{2\pi}{8} \times 4} & e^{-j\frac{2\pi}{8} \times 6} \\ 1 & e^{-j\frac{2\pi}{8} \times 3} & e^{-j\frac{2\pi}{8} \times 6} & e^{-j\frac{2\pi}{8} \times 1} & e^{-j\frac{2\pi}{8} \times 4} & e^{-j\frac{2\pi}{8} \times 7} & e^{-j\frac{2\pi}{8} \times 2} & e^{-j\frac{2\pi}{8} \times 5} \\ 1 & e^{-j\frac{2\pi}{8} \times 4} & e^{-j\frac{2\pi}{8} \times 0} & e^{-j\frac{2\pi}{8} \times 4} & e^{-j\frac{2\pi}{8} \times 0} & e^{-j\frac{2\pi}{8} \times 4} & e^{-j\frac{2\pi}{8} \times 0} & e^{-j\frac{2\pi}{8} \times 4} \\ 1 & e^{-j\frac{2\pi}{8} \times 5} & e^{-j\frac{2\pi}{8} \times 2} & e^{-j\frac{2\pi}{8} \times 7} & e^{-j\frac{2\pi}{8} \times 4} & e^{-j\frac{2\pi}{8} \times 1} & e^{-j\frac{2\pi}{8} \times 6} & e^{-j\frac{2\pi}{8} \times 3} \\ 1 & e^{-j\frac{2\pi}{8} \times 6} & e^{-j\frac{2\pi}{8} \times 4} & e^{-j\frac{2\pi}{8} \times 2} & e^{-j\frac{2\pi}{8} \times 0} & e^{-j\frac{2\pi}{8} \times 6} & e^{-j\frac{2\pi}{8} \times 4} & e^{-j\frac{2\pi}{8} \times 2} \\ 1 & e^{-j\frac{2\pi}{8} \times 7} & e^{-j\frac{2\pi}{8} \times 6} & e^{-j\frac{2\pi}{8} \times 5} & e^{-j\frac{2\pi}{8} \times 4} & e^{-j\frac{2\pi}{8} \times 3} & e^{-j\frac{2\pi}{8} \times 2} & e^{-j\frac{2\pi}{8} \times 1} \end{pmatrix}$$

2.31 用DFT进行图像展开时的基图像是怎样的

因为DFT的核是复函数, 它的基图像自然是复的, 有实部和虚部。并且可由矩阵 U 的任意两行的外积得到。基图像的实部如图2-7所示, 虚部如图2-8所示, 其中矩阵 U 从例2.27中计算而得。

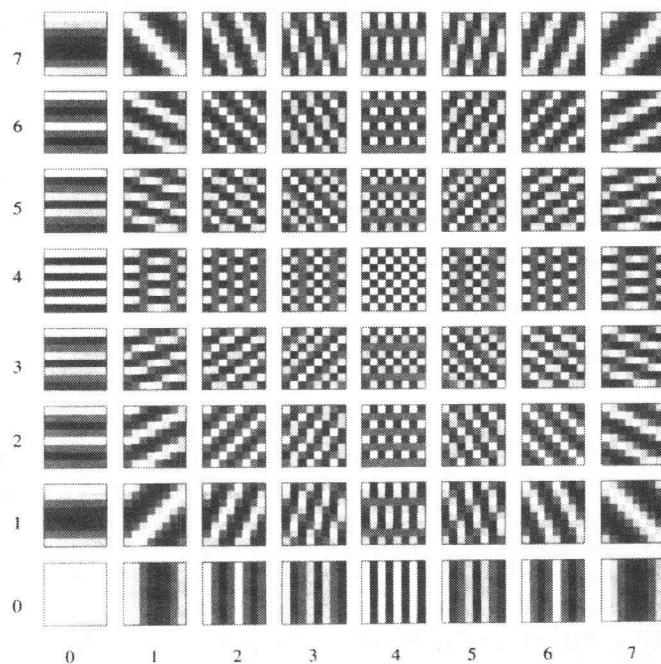


图2-7 傅里叶变换基图像的实部

每个基图像的值已经线性量化到0（黑色）至255（白色）的范围内。上图左边和底部的数字表示相应位置的图像是由矩阵 U 的哪些行相乘得到。

68

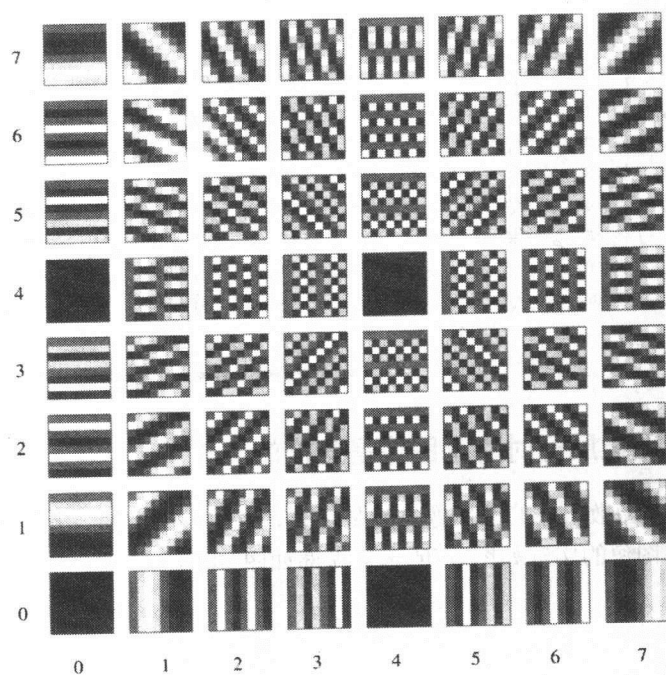


图2-8 傅里叶变换基图像的虚部

• 例2.28 计算下面图像的离散傅里叶变换的实部和虚部:

$$g = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

应用例2.25中矩阵 U

$$\begin{aligned} A &= UgU \\ gU &= \frac{1}{2} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 2 & -1-j & 0 & j-1 \\ 2 & -1-j & 0 & j-1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ A &= \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 2 & -1-j & 0 & j-1 \\ 2 & -1-j & 0 & j-1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ &= \frac{1}{4} \begin{pmatrix} 4 & -2-2j & 0 & 2j-2 \\ -2j-2 & 2j & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 2j-2 & 2 & 0 & -2j \end{pmatrix} \\ &= \begin{pmatrix} 1 & -\frac{1+j}{2} & 0 & \frac{j-1}{2} \\ -\frac{1+j}{2} & \frac{j}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 \\ \frac{j-1}{2} & \frac{1}{2} & 0 & -\frac{j}{2} \end{pmatrix} \\ Re(A) &= \begin{pmatrix} 1 & -\frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 \\ -\frac{1}{2} & \frac{1}{2} & 0 & 0 \end{pmatrix} \quad \text{和} \quad Im(A) = \begin{pmatrix} 0 & -\frac{1}{2} & 0 & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & -\frac{1}{2} \end{pmatrix} \end{aligned}$$

69

例2.29 写出例2.14中图像的傅里叶变换过程。

图2-9所示的8个图像均是重构的图像, 分别使用了 U 矩阵的1、2、……、8行进行重构。
重构图像的平方误差和分别如下:

图2-9a的误差: 366394

图2-9b的误差: 285895

图2-9c的误差: 234539

图2-9d的误差: 189508

图2-9e的误差: 141481

图2-9f的误差: 119612

图2-9g的误差: 71908

图2-9h的误差: 0

70

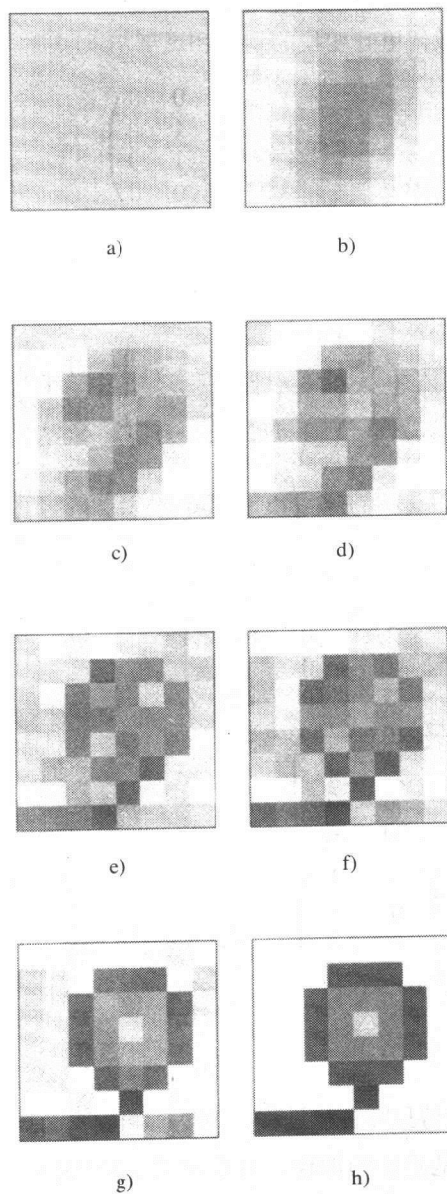


图2-9 重构后的图像

所使用的基图像从左上到右下，分别由例2.27中矩阵 U 的前1、2、3、……、8个行产生。

71

2.32 为什么离散傅里叶变换比其他变换使用更普遍

与沃尔什变换相比，离散傅里叶变换最主要的优点在于它的定义形式满足卷积定理。虽然沃尔什函数也能导出类似的定理，但沃尔什变换与卷积之间的关系不如傅里叶变换来得简

单, 另外在计算机上实现的开销也不小。卷积定理使傅里叶变换成为迄今为止在图像处理方面最具吸引力的变换。

此外, 傅里叶变换的基函数有很好的表达细节能力, 能比其他变换以更小的误差接近原图, 而其他变换由于保留了固定个数的项使误差变大。比较例2.29与例2.23、例2.42的重构误差, 可以得出上述结论。我们还必须比较重构后的图2-9a、图2-9b和图2-9d的误差, 它们分别保留了前 2^0 、 2^1 以及 2^2 个基函数。

另一方面, 注意当我们保留 n 个基图像时, 傅里叶变换中实际需要 $2n$ 个系数来重构, 而哈尔变换和沃尔什变换仅仅需要 n 个系数。这是由于傅里叶系数是复数, 实部和虚部都需要存储或传送。

2.33 什么是卷积定理

卷积定理: 两个函数卷积的傅里叶变换, 等于它们各自傅里叶变换的乘积。如果函数是在有限维空间中定义的图像, 只有假设每个图像在各个方向上都周期性地重复, 卷积定理才成立。

B2.4 如果一个函数是另外两个函数的卷积, 那么它的DFT与那两个函数的DFT有什么关系?

假设对两个2维离散函数 $g(n, m)$ 和 $w(n, m)$ 求卷积, 得到另一个函数 $v(n, m)$:

$$v(n, m) = \sum_{n'=0}^{N-1} \sum_{m'=0}^{M-1} g(n-n', m-m')w(n', m') \quad (2-49)$$

记这三个函数的离散傅里叶变换分别为 $\hat{v}, \hat{g}, \hat{w}$ 。为了找出三者的关系, 计算 $v(n, m)$ 的DFT。为此, 在式(2-49)两边乘以下面的核

$$\frac{1}{\sqrt{NM}} \exp \left[-j2\pi \left(\frac{pn}{N} + \frac{qm}{M} \right) \right] \quad (2-50)$$

并对所有的 m 和 n 求和, 式(2-49)则变成:

$$\begin{aligned} & \frac{1}{\sqrt{NM}} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} v(n, m) e^{-2\pi j \left[\frac{pn}{N} + \frac{qm}{M} \right]} \\ &= \frac{1}{\sqrt{NM}} \sum_{n'=0}^{N-1} \sum_{m'=0}^{M-1} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} g(n-n', m-m')w(n', m') e^{-2\pi j \left[\frac{pn}{N} + \frac{qm}{M} \right]} \end{aligned} \quad (2-51)$$

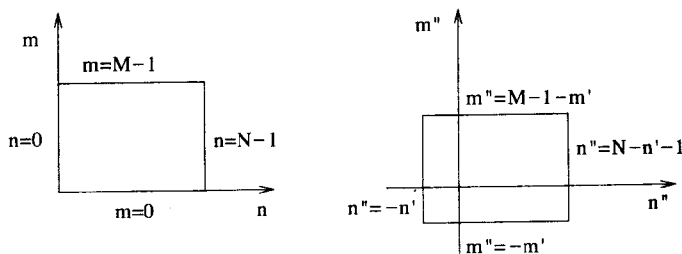
可以认为, 该表达式左边是 v 的离散傅里叶变换; 也就是

$$\hat{v}(p, q) = \frac{1}{\sqrt{NM}} \sum_{n'=0}^{N-1} \sum_{m'=0}^{M-1} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} g(n-n', m-m')w(n', m') e^{-2\pi j \frac{pn}{N}} e^{-2\pi j \frac{qm}{M}}$$

下面把它分为两个连加和式的乘积。为此, g, w 的下标必须相互独立。引进新的记号:

$$n-n' \equiv n'', \quad m-m' \equiv m''$$

即 $n = n' + n''$, $m = m' + m''$ 。并找出 m'' 和 n'' 的变化范围。把 (n, m) 空间中求和的区域映射到对应的 (n'', m'') 空间上:



在 (n, m) 空间中, 求和的区域被四条直线包围, 下列左边的等式给出了直线方程。把每一条都变换成 (n'', m'') 空间的另一直线, 并在右边给出相应的直线方程。经过变换的直线定义了新的求和范围。

$$\begin{aligned} m = 0 &\rightarrow m'' = -m' \\ m = M - 1 &\rightarrow m'' = M - m' - 1 \\ n = 0 &\rightarrow n'' = -n' \\ n = N - 1 &\rightarrow n'' = N - n' - 1 \end{aligned}$$

$\hat{v}(p, q)$ 最后的表达式变为:

$$\begin{aligned} \hat{v}(p, q) = & \frac{1}{\sqrt{NM}} \sum_{n'=0}^{N-1} \sum_{m'=0}^{M-1} w(n', m') e^{-2\pi j [\frac{pn'}{N} + \frac{qm'}{M}]} \\ & \sum_{m''=m'}^{M-1-m'} \sum_{n''=n'}^{N-1-n'} g(n'', m'') e^{-2\pi j [\frac{pn''}{N} + \frac{qm''}{M}]} \end{aligned} \quad (2-52)$$

注意式(2-52)的最后一行, 把对 m'' , n'' 两个变量求和的部分记为因式 T 。把正负标号分离, 改写为:

$$T \equiv \left[\sum_{m''=-m'}^{-1} + \sum_{m''=0}^{M-1-m'} \right] \left[\sum_{n''=-n'}^{-1} + \sum_{n''=0}^{N-1-n'} \right] g(n'', m'') e^{-2\pi j [\frac{pn''}{N} + \frac{qm''}{M}]} \quad (2-53)$$

显然, 两个数组 g 和 w 都没有对于负标号的定义。将 g 和 w 的定义延拓到 $[0, N-1]$, $[0, M-1]$ 范围以外, 使之符合我们的要求。检验因式:

$$\sum_{m''=-m'}^{-1} g(n'', m'') e^{-2\pi j q \frac{m''}{M}}$$

定义新的变量 $m''' = M + m''$ 。上述表达式改写为

$$\sum_{m'''=M-m'}^{M-1} g(n'', -M + m''') e^{-2\pi j q \frac{m'''}{M}} e^{-2\pi j q}$$

至此, 如果选择定义: $g(n'', m''' - M) \equiv g(n'', m''')$, 则上面和式可以改写为:

$$\sum_{m'''=M-m'}^{M-1} g(n'', m''') e^{-2\pi j q \frac{m'''}{M}}, \quad \text{由于 } e^{-2\pi j q} = 1$$

m''' 仅仅是一个哑标号, 不具实际意义, 可改写为其他任意形式。现在把它改写为 m'' 。上述表达式变为:

$$\sum_{m''=M-m'}^{M-1} g(n'', m'') e^{-2\pi j q \frac{m''}{M}}$$

把上面的结果和下面的项相加:

$$\sum_{m''=0}^{M-m'-1} g(n'', m'') e^{-2\pi j q \frac{m''}{M}}$$

和式写为:

$$\sum_{m''=0}^{M-1} g(n'', m'') e^{-2\pi j q \frac{m''}{M}}$$

对 n'' 做相同的处理。假设 g 是对于变量 n'' 周期为 N 的周期函数。在这样的假设下, 延拓 g 原来的定义域, 式(2-53)中定义的因式 T 就是:

$$T = \sum_{m''=0}^{M-1} \sum_{n''=0}^{N-1} g(n'', m'') e^{-2\pi j [\frac{p n''}{N} + \frac{q m''}{M}]} \quad (2-54)$$

上式不包含指标 m' , n' , 因式 T 还要乘上另一和式, 也就是式(2-52)中对 n' , m' 求和的部分。 T 实际上等于 $\sqrt{MN} \hat{g}(p, q)$ 。同样地, 式(2-52)中对 $n'm'$ 求和的部分实际上等于 w 的离散傅里叶变换。因此式(2-52)变为:

$$\hat{v}(p, q) = \sqrt{MN} \hat{g}(p, q) \hat{w}(p, q) \quad (2-55)$$

在以下假设下成立:

$$\begin{aligned} g(n, m) &\equiv g(n - N, m - M) \\ w(n, m) &\equiv w(n - N, m - M) \\ g(n, m) &\equiv g(n, m - M) \\ w(n, m) &\equiv w(n, m - M) \\ g(n, m) &\equiv g(n - N, m) \\ w(n, m) &\equiv w(n - N, m) \end{aligned}$$

假定图像阵列 g 和 m 在空间 (n, m) 的两个方向上都具有周期性, 周期分别为 M 和 N 。对应于时域上的卷积定理 (time convolution theorem)。

频域上的卷积定理 (frequency convolution theorem) 正好也有相同的形式。由于离散傅里叶变换和它的逆之间有对称关系, 表明了这些函数的离散傅里叶变换在整个 (q, p) 空间上也具有周期性, 并分别以 N, M 为周期。

注意: 式(2-55)中存在系数因子 \sqrt{MN} 的原因在于, 如此定义的离散傅里叶变换形式使得其正、逆变换是完全对称的。

75

- 例2.30(B) $g(n, m)$ 和 $w(n, m)$ 是两个 $N \times M$ 的图像。它们的DFT分别为 $\hat{g}(p, q)$ 和 $\hat{w}(p, q)$ 。以如下方式产生图像 $x(n, m)$

$$x(n, m) = g(n, m) \times w(n, m) \quad (2-56)$$

利用 $\hat{g}(p, q)$ 和 $\hat{w}(p, q)$ 来表示 $x(n, m)$ 的DFT $\hat{x}(p, q)$ 。

对式(2-56)两边进行DFT:

$$\hat{x}(k, l) = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} g(n, m) w(n, m) e^{-2\pi j \left[\frac{kn}{M} + \frac{ln}{N} \right]}$$

以 \hat{g}, \hat{w} 代入 g 和 w :

$$\begin{aligned} \hat{x}(k, l) &= \frac{1}{\sqrt{MN}} \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{q=0}^{M-1} \sum_{p=0}^{N-1} \hat{g}(p, q) e^{-2\pi j \left[\frac{pn}{M} + \frac{qn}{N} \right]} \\ &\quad \sum_{r=0}^{M-1} \sum_{s=0}^{N-1} \hat{w}(s, r) e^{-2\pi j \left[\frac{rs}{M} + \frac{rs}{N} \right]} e^{-2\pi j \left[\frac{kn}{M} + \frac{ln}{N} \right]} = \\ &= \frac{1}{\sqrt{MN}} \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{q=0}^{M-1} \sum_{p=0}^{N-1} \sum_{r=0}^{M-1} \sum_{s=0}^{N-1} \hat{g}(p, q) \hat{w}(s, r) e^{2\pi j \left[\frac{n(s+p)}{N} + \frac{m(r+q)}{M} \right]} e^{-2\pi j \left[\frac{kn}{M} + \frac{ln}{N} \right]} \end{aligned}$$

引入新的变量 u 和 v , 代替 s 和 r :

$$\begin{aligned} u &= r + q \rightarrow u \text{ 的取值范围 } [q, M-1+q] \\ v &= p + s \rightarrow v \text{ 的取值范围 } [p, N-1+p] \end{aligned}$$

因此

$$\begin{aligned} \hat{x}(k, l) &= \frac{1}{\sqrt{MN}} \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{q=0}^{M-1} \sum_{p=0}^{N-1} \sum_{u=q}^{M-1+q} \sum_{v=p}^{N-1+p} \hat{g}(p, q) \\ &\quad \hat{w}(v-p, u-q) e^{-2\pi j \left[\frac{nv}{N} + \frac{mu}{M} \right]} e^{-2\pi j \left[\frac{kn}{M} + \frac{ln}{N} \right]} \\ \Rightarrow \hat{x}(k, l) &= \frac{1}{\sqrt{MN}} \frac{1}{MN} \sum_{q=0}^{M-1} \sum_{p=0}^{N-1} \sum_{u=q}^{M-1+q} \sum_{v=p}^{N-1+p} \hat{g}(p, q) \\ &\quad \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} e^{2\pi j \left[\frac{m(u-k)}{M} + \frac{n(v-l)}{N} \right]} \end{aligned}$$

76

则 (参见式(2-43)):

$$\sum_{m=0}^{M-1} e^{2\pi j \frac{m}{M} t} \begin{cases} M & t=0 \\ 0 & t \neq 0 \end{cases}$$

因此

$$\begin{aligned} \hat{x}(k, l) &= \frac{1}{\sqrt{MN}} \frac{1}{MN} \sum_{q=0}^{M-1} \sum_{p=0}^{N-1} \sum_{u=q}^{M-1+q} \sum_{v=p}^{N-1+p} \hat{g}(p, q) \\ &\quad \hat{w}(v-p, u-q) M \delta(u-k) N \delta(v-l) \\ &= \underbrace{\frac{1}{\sqrt{MN}} \sum_{q=0}^{M-1} \sum_{p=0}^{N-1} \hat{g}(p, q) \hat{w}(k-p, l-q)}_{\hat{g} \text{ 和 } \hat{w} \text{ 的卷积}} \end{aligned}$$

例2.31 求证: 如果 $g(k, l)$ 是 $M \times N$ 的图像, 定义为整个 (k, l) 空间上分别以 N, M 为周期的周期函数。则它的DFT $\hat{g}(m, n)$ 在 (m, n) 空间上也是周期性的, 并且具有相同周期。

要证明 $\hat{g}(m+N, n+N) = \hat{g}(m, n)$ 。从 $\hat{g}(m, n)$ 的定义入手:

$$\hat{g}(m, n) = \frac{1}{\sqrt{MN}} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} g(k, l) e^{-2\pi j \left[\frac{km}{M} + \frac{ln}{N} \right]}$$

则

$$\begin{aligned} \hat{g}(m+M, n+N) &= \frac{1}{\sqrt{MN}} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} g(k, l) e^{-2\pi j \left[\frac{k(m+M)}{M} + \frac{l(n+N)}{N} \right]} \\ &= \frac{1}{\sqrt{MN}} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} g(k, l) e^{-2\pi j k} e^{-2\pi j l} = \hat{g}(m, n) \end{aligned}$$

77

例2.32(B) 求证: 如果 $v(n, m)$ 定义如下:

$$v(n, m) \equiv \sum_{n'=0}^{N-1} \sum_{m'=0}^{M-1} g(n-n', m-m') w(n', m') \quad (2-57)$$

其中, $g(n, m)$ 和 $w(n, m)$ 是两个具有周期的图像函数, 两个变量均分别以 N, M 为周期。则 $v(n, m)$ 可以写为:

$$v(n, m) = \sum_{n'=0}^{N-1} \sum_{m'=0}^{M-1} w(n-n', m-m') g(n', m') \quad (2-58)$$

证明: 定义新的变量:

$$\begin{aligned} n-n' &\equiv k \Rightarrow n' = n-k \\ m-m' &\equiv l \Rightarrow m' = m-l \end{aligned}$$

则式(2-57)变为

$$\Rightarrow v(n, m) = \sum_{k=n}^{n-N+1} \sum_{l=m}^{m-M+1} g(k, l)w(n-k, m-l) \quad (5-59)$$

考虑下面的和式:

$$\begin{aligned} \sum_{k=n}^{n-N+1} g(k, l)w(n-k, m-l) &= \sum_{k=-N+n+1}^n g(k, l)w(n-k, m-l) \\ &= \underbrace{\sum_{k=-N}^{-1} g(k, l)w(n-k, m-l)}_{\text{变量替换 } \tilde{k}=k+N} - \underbrace{\sum_{k=-N}^{-N+n} g(k, l)w(n-k, m-l)}_{\text{变量替换 } \tilde{k}=k+N} \\ &\quad + \sum_{k=0}^n g(k, l)w(n-k, m-l) \\ &= \sum_{\tilde{k}=0}^{N-1} g(\tilde{k}-N, l)w(n-\tilde{k}+N, m-l) - \sum_{\tilde{k}=0}^n g(\tilde{k}-N, l)w(n-\tilde{k}+N, m-l) \\ &\quad + \sum_{k=0}^n g(k, l)w(n-k, m-l) \end{aligned}$$

$$\text{由于 } g \text{ 以 } N \text{ 周期} \Rightarrow g(k-N, l) = g(k, l)$$

$$\text{由于 } w \text{ 以 } N \text{ 周期} \Rightarrow w(s+N, t) = w(s, t)$$

因此, 最后两个和式恒等, 可以相互抵消。式(2-59)是对 k 从0到 $N-1$ 求和。同样地, 可以证明式(2-59)也是对 l 从0到 $M-1$ 求和。因此式(2-58)成立。

...

2.34 如何显示图像的离散傅里叶变换

假设 $\hat{g}(p, q)$ 是一幅图像的离散傅里叶变换。 $\hat{g}(p, q)$ 中的量是图像展开成离散傅里叶函数时的系数, 每个对应二维空间 (k, l) 上的一对不同的频率。当 p, q 增大时, 高频信息对图像的贡献变得越来越次要。因此相应的 $\hat{g}(p, q)$ 系数值变小。因为系数值的变化范围较大使显示这些系数变得困难。因此为了显示的这个频谱值, 引进对数函数:

$$d(p, q) \equiv \log(1 + |\hat{g}(p, q)|).$$

函数 d 把 $\hat{g}(p, q)$ 按比例缩放到可显示的灰度值范围内, $d(p, q)$ 可以作为 $\hat{g}(p, q)$ 的替代显示。注意当 $\hat{g}(p, q) = 0$ 时, $d(p, q) = 0$ 。函数 $d(p, q)$ 有以下性质: 降低较大和较小函数值之间的比值, 使得较大和较小函数值都能在同一比例下显示。例如, 如果 $\hat{g}_{\max} = 10$, $\hat{g}_{\min} = 0.1$, 那么在同一坐标系中把 g 的取值范围精确完整地显示显然比较困难, 因为它们的比值为100。考虑 $\log(11) = 1.041$ 和 $\log(1.1) = 0.041$, 它们的比值仅为25。因此可以把两个数在同一坐标下简单地显示出来。

2.35 图像旋转时的离散傅里叶变换是怎样的

改写离散傅里叶变换的定义, 也就是式(2-37)在正方形图像的情况下的特殊形式 (令 $M = N$):

$$\hat{g}(m, n) = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} g(k, l) e^{-2\pi j \frac{km+ln}{N}} \quad (2-60)$$

对于平面 (k, l) 和 (m, n) , 引进极坐标如下: $k = r \cos \theta, l = r \sin \theta, m = \omega \cos \phi, n = \omega \sin \phi$ 。注意, $km + ln = r\omega(\cos \theta \cos \phi + \sin \theta \sin \phi) = r\omega \cos(\theta - \phi)$ 。式(2-60)变为:

$$\hat{g}(\omega, \phi) = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} g(r, \theta) e^{-2\pi j \frac{r\omega \cos(\theta - \phi)}{N}} \quad (2-61) \quad \boxed{79}$$

求和时的变量 k, l 虽然没有显式地出现在被加数中, 却是隐含存在的, 并且求和过程应该遍历所有相关的点。从 k, l 的值能得到对应 r, θ 的值。

假设对 $g(r, \theta)$ 旋转一个角度 θ_0 , 变成 $g(r, \theta + \theta_0)$ 。下面求旋转后的函数的离散傅里叶变换。式(2-61)也是一个“带插槽的变换机”。把函数“插进”适当的变换位置, 马上能输出它的 DFT。因此, 应用式(2-61)计算 $g(r, \theta + \theta_0)$ 的 DFT, 只需简单地把 $g(r, \theta)$ 换成 $g(r, \theta + \theta_0)$ 。把 $g(r, \theta + \theta_0)$ 的 DFT 记为 $\hat{\hat{g}}(\omega, \phi)$ 。得到:

$$\hat{\hat{g}}(\omega, \phi) = \frac{1}{N} \underbrace{\sum \sum}_{\text{所有点}} g(r, \theta + \theta_0) e^{-2\pi j \frac{r\omega \cos(\theta - \phi)}{N}} \quad (2-62)$$

要找出 $\hat{\hat{g}}(\omega, \phi)$ 与 $\hat{g}(\omega, \phi)$ 之间的关系, 必须设法使 $g(r, \theta)$ 出现在式(2-62)的右边。为此, 在式(2-62)中引进新的变量, $\bar{\theta} \equiv \theta + \theta_0$:

$$\hat{\hat{g}}(\omega, \phi) = \frac{1}{N} \underbrace{\sum \sum}_{\text{所有点}} g(r, \bar{\theta}) e^{-2\pi j \frac{r\omega \cos(\bar{\theta} - \theta_0 - \phi)}{N}} \quad (2-63)$$

我们认为, 式(2-63)的右边是旋转前的图像 $g(r, \theta)$ 的 DFT 在 $\phi + \theta_0$ 处的值, 而不是原来 ϕ 处的值, 也就是说把等式右边看做 $\hat{g}(\omega, \phi + \theta_0)$ 。所以有:

$$\hat{\hat{g}}(\omega, \phi) = \hat{g}(\omega, \phi + \theta_0) \quad (2-64)$$

旋转了 θ_0 的图像的 DFT = 未旋转图像的 DFT 旋转 θ_0 。

- 例2.33 将例2.26中的图像顺时针旋转 90° , 重新计算它的离散傅里叶变换。从而检验2D图像的离散傅里叶变换与原图旋转 θ_0 后的离散傅里叶变换之间的关系。

旋转 90° 后的图像:

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

80

为了计算它的DFT, 首先右乘例2.25中的矩阵 U , 然后左乘相同的矩阵 U :

$$\begin{aligned} & \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ & \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix} \frac{1}{2} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ & = \frac{1}{4} \begin{pmatrix} 4 & 0 & 0 & 0 \\ -4 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ -4 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix} \end{aligned}$$

对比上述结果与例2.26的结果, 可以发现图像先经过 90° 旋转再做离散傅里叶变换, 等于将旋转前图像先做离散傅里叶变换, 再进行 90° 旋转。

•
•
•

2.36 图像平移时的离散傅里叶变换是怎样的

假设将图像平移到点 (k_0, l_0) , 变成 $g(k-k_0, l-l_0)$ 。计算平移后图像的DFT, 应用式(2-60)。记 $g(k-k_0, l-l_0)$ 的DFT为 $\hat{g}(m, n)$, 得到:

$$\hat{g}(m, n) = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} g(k-k_0, l-l_0) e^{-2\pi j \frac{km+ln}{N}} \quad (2-65)$$

为了找出 $\hat{g}(m, n)$ 和 $\hat{g}(m, n)$ 的关系, 必须设法使 $g(k, l)$ 出现在上式的右边。为此, 定义新的变量 $k' \equiv k-k_0, l' \equiv l-l_0$, 有:

$$\hat{g}(m, n) = \frac{1}{N} \sum_{k'=-k_0}^{N-1-k_0} \sum_{l'=-l_0}^{N-1-l_0} g(k', l') e^{-2\pi j \frac{k'm+l'n}{N}} e^{-2\pi j \frac{k_0m+l_0n}{N}} \quad (2-66)$$

由于假设图像在各个方向上都周期性重复, 所以求和时这些方向并不重要, 只要求和窗口的大小适当便可以。换句话说, 无论求和指标 k', l' 取什么初值, 只要求和遍 N 个连续的值。因此假设式(2-66)中 k', l' 取值从0到 $N-1$ 。注意, 因式 $e^{-2\pi j \frac{k_0m+l_0n}{N}}$ 与 k', l' 无关。因此可以提到等式外面。可以发现式(2-66)右边出现了 $g(k, l)$ 的DFT。(注意 k', l' 仅仅是哑指标, 不具有实际意义, 无论把它记为 k', l' 或者 k, l 都是没有区别的) 因此有:

81

$$\hat{g}(m, n) = \hat{g}(m, n) e^{-2\pi j \frac{k_0 m + l_0 n}{N}} \quad (2-67)$$

$$\text{平移图像的DFT} = \text{原图的DFT} \times e^{-2\pi j \frac{k_0 m + l_0 n}{N}}$$

类似地, 可以证明:

$$\text{图像经平移的DFT} = (\text{原图} \times e^{2\pi j \frac{m_0 k + n_0 l}{N}}) \text{的DFT}$$

或者

$$\hat{g}(m-m_0, n-n_0) = (\text{原图} \times e^{2\pi j \frac{m_0 k + n_0 l}{N}}) \text{的DFT}$$

2.37 函数平均值和它的DFT之间的关系是怎样的

函数平均值定义如下:

$$\bar{g} = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} g(k, l) \quad (2-68)$$

式(2-60)中如果令 $m = n = 0$, 有

$$\hat{g}(0, 0) = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} g(k, l) \quad (2-69)$$

所以, 图像的平均值与图像DFT的直流分量 (direct component, dc) 有以下关系:

$$\bar{g} = \frac{1}{N} \hat{g}(0, 0) \quad (2-70)$$

• 例2.34 验证图像平均值

$$g = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

82

与图像的离散傅里叶变换的关系。

应用离散傅里叶变换的公式(式2-37), 令 $N=M=4$, $n=m=0$:

$$\begin{aligned} \hat{g}(0, 0) &= \frac{1}{4} \sum_{k=0}^3 \sum_{l=0}^3 g(k, l) = \frac{1}{4} (0+0+0+0+0+1+1+0+0+1+1+0 \\ &\quad +0+0+0+0) = 1 \end{aligned}$$

g 的平均值:

$$\begin{aligned} \bar{g} &= \frac{1}{16} \sum_{k=0}^3 \sum_{l=0}^3 g(k, l) = \frac{1}{16} (0+0+0+0+0+1+1+0+0+1+1+0 \\ &\quad +0+0+0+0) = \frac{4}{16} = \frac{1}{4} \end{aligned}$$

从而 $N\bar{g} = 4 \times \frac{1}{4} = 1$, 式(2-70)成立。

2.38 图像伸缩时的离散傅里叶变换是怎样的

利用离散函数的特点, 在其定义域范围内的一个区间上, 隐式地进行以下操作: 把区间分为更小的区域, 每个大小为 $\Delta x \times \Delta y$, 取函数在各个小区域中心的函数值作为函数在整个小区域上的值。把所有函数值相加, 再除以小区域的个数。这样, 函数平均值定义为:

$$\bar{g} = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} g(x, y) \Delta x \Delta y \quad (2-71)$$

因为 x, y 均以 1 为步长, 为简单起见, 设 $\Delta x = 1, \Delta y = 1$ 。注意, 从离散傅里叶变换的定义可以看出它是一个加权平均值, 其中 $g(k, l)$ 在每个小区域上都乘以不同的权重。这样分析 DFT 的话, 正确的离散傅里叶变换的定义应该包含 $\Delta k \times \Delta l$ 因子。且 g 在每个小区域上是常量。之所以忽略了 $\Delta k \times \Delta l$ 因子, 是因为 $\Delta k = \Delta l = 1$ 。DFT 的准确描述如下:

$$\hat{g}(m, n) = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} g(k, l) e^{-2\pi j \left(\frac{km+ln}{N} \right)} \Delta k \Delta l \quad (2-72)$$

假设在 (k, l) 平面上, 改变 $g(k, l)$ 的尺度, 变成 $g(\alpha k, \beta l)$ 。把伸缩后 g 的 DFT 记为 $\hat{g}(m, n)$ 。将 $g(\alpha k, \beta l)$ 代替式(2-72)的 $g(k, l)$ 。则:

$$\hat{g}(m, n) = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} g(\alpha k, \beta l) e^{-2\pi j \frac{km+ln}{N}} \Delta k \Delta l \quad (2-73)$$

为了找出 $\hat{g}(m, n)$ 和 $\hat{g}(m, n)$ 之间的关系, 必须设法使 $g(k, l)$ 出现在式(2-73)的右边。为此, 定义新的变量 $k' \equiv \alpha k, l' \equiv \beta l$ 。则:

$$\hat{g}(m, n) = \frac{1}{N} \sum_{k'=0}^{\alpha(N-1)} \sum_{l'=0}^{\beta(N-1)} g(k', l') e^{-2\pi j \frac{k' \frac{m}{\alpha} + l' \frac{n}{\beta}}{N}} \frac{\Delta k'}{\alpha} \frac{\Delta l'}{\beta} \quad (2-74)$$

除非求和变量 k', l' 不是以步长 1 增长, 否则上式的求和过程能遍历函数 $g(k', l')$ 全部有定义的点。再次把式(2-74)右边看做 $g(k, l)$, 只不过现在是在点 $(\frac{m}{\alpha}, \frac{n}{\beta})$ 上计算, 而不是原来的 (m, n) 。可以写成:

$$\hat{g}(m, n) = \frac{1}{\alpha\beta} \hat{g}\left(\frac{m}{\alpha}, \frac{n}{\beta}\right) \quad (2-75)$$

伸缩函数的 DFT = $\frac{1}{|\text{伸缩因子的乘积}|} \times \text{未伸缩的函数的 DFT 在相同的}$

点上反方向伸缩。

B2.5 什么是快速傅里叶变换?

到现在为止, 我们所处理的变换都是可分离的。这意味着计算两个一维变换与计算一个二维变换不同。而计算二维离散傅里叶变换时却可以化为两个一维离散傅里叶变换来计算, 这样就可以使用专门的算法来提高速度和效率。这里将给出快速傅里叶变换算法的简单描述, 称为逐次加倍算法 (successive doubling algorithm)。下面以一维为例进行讨论。离散傅里叶变换定义为:

$$\hat{f}(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) w_N^{ux} \quad (2-76)$$

其中 $w_N = e^{-\frac{2\pi j}{N}}$ 。设 $N = 2^n$, 把 N 改写成 $2M$, 代入上式:

$$\hat{f}(u) = \frac{1}{2M} \sum_{x=0}^{2M-1} f(x) w_{2M}^{ux} \quad (2-77)$$

分离 f 的变量在奇数点的值和偶数点的值。则:

$$\hat{f}(u) = \frac{1}{2} \left\{ \frac{1}{M} \sum_{x=0}^{M-1} f(2x) w_{2M}^{u(2x)} + \frac{1}{M} \sum_{x=0}^{M-1} f(2x+1) w_{2M}^{u(2x+1)} \right\} \quad (2-78)$$

显然 $w_{2M}^{2ux} = w_M^{ux}$, $w_{2M}^{2ux+u} = w_M^{ux} w_{2M}^u$ 。则:

$$\hat{f}(u) = \frac{1}{2} \left\{ \frac{1}{M} \sum_{x=0}^{M-1} f(2x) w_M^{ux} + \frac{1}{M} \sum_{x=0}^{M-1} f(2x+1) w_M^{ux} w_{2M}^u \right\} \quad (2-79)$$

可以写成:

$$\hat{f}(u) \equiv \frac{1}{2} \left\{ \hat{f}_{\text{even}}(u) + \hat{f}_{\text{odd}}(u) w_{2M}^u \right\} \quad (2-80)$$

其中 $\hat{f}_{\text{even}}(u)$ 定义为函数 f 的偶取样点的 DFT, $\hat{f}_{\text{odd}}(u)$ 定义为函数 f 的奇取样点的 DFT。但是式(2-80)中 $\hat{f}(u)$ 仅在 $u < M$ 时有定义。必须定义 $\hat{f}(u)$ 在 $u = 0, 1, \dots, N$ 上的值, 也就是 u 取值从 0 直到 $2M$ 。为此应用式(2-79) \hat{f} 的变量变为 $u + M$:

$$\hat{f}(u + M) = \frac{1}{2} \left\{ \frac{1}{M} \sum_{x=0}^{M-1} f(2x) w_M^{u+Mx} + \frac{1}{M} \sum_{x=0}^{M-1} f(2x+1) w_M^{u+Mx} w_{2M}^{u+M} \right\} \quad (2-81)$$

然而

$$\begin{aligned} w_M^{u+M} &= w_M^u w_M^M = w_M^u \\ w_{2M}^{u+M} &= w_{2M}^u w_{2M}^M = -w_{2M}^u \end{aligned} \quad (2-82)$$

所以

$$\hat{f}(u+M) = \frac{1}{2} \left\{ \hat{f}_{\text{even}}(u) - \hat{f}_{\text{odd}}(u) w_{2M}^u \right\} \quad (2-83)$$

其中

$$\hat{f}_{\text{even}}(u) \equiv \frac{1}{M} \sum_{x=0}^{M-1} f(2x) w_M^{ux}, \quad \hat{f}_{\text{odd}}(u) \equiv \frac{1}{M} \sum_{x=0}^{M-1} f(2x+1) w_M^{ux} \quad (2-84)$$

注意式(2-80)、式(2-83)以及定义式(2-84)合起来就是 $\hat{f}(u)$ 的完整定义。因此一个 N 点的变换可以利用式(2-84)通过计算两次 $\frac{N}{2}$ 个点的变换得到。然后式(2-80)、式(2-83)可以用于计算整个变换。可以证明, 该算法复杂性从原来的 N^2 降到 Nn ; 也就是 $N \log_2 N$ 。这也就是为什么图像的维数是2的幂的原因。

2.39 什么是离散余弦变换

如果矩阵 h_c, h_r 的行是某类切比雪夫多项式 (Chebyshev polynomial) 的离散形式, 我们有偶对称余弦变换 (even symmetrical cosine transform)。之所以如此命名, 是因为它等同于假定 $2N \times 2N$ 的图像是由一个 $N \times N$ 图像沿两边对称延拓而成的。再对该对称图像施行傅里叶变换时, 就是真正的余弦变换。如此定义的离散余弦变换 (discrete cosine transform, DCT) 在 JPEG 编码时得到广泛应用, 具体做法是先把每幅图像分解为 8×8 的小块, 再进行分块的 DCT 变换, 最后将变换后的 DCT 系数进行编码后传输。

2.40 本章要点

本章介绍了图像处理中应用的线性变换、酉变换和可分离变换。这些变换把图像分解为一系列基图像的线性组合。通常情况下, 这些基图像在结构 (细节) 上以升序排列。使得我们可以按照不同需要, 从第一个基图像开始, 可以用任意多的基图像以任意精度来表示一幅图像。如果使用图像本身定义的基图像, 能使重构达到最理想的效果。但是由于如此构造的基图像与原图本身相关, 这种做法并不高效。作为另一种选择, 标准正交函数集能建立一些预定义的基图像。它们能捕捉所有图像的基本特征。一旦确定正交函数集后, 在图像通信时, 只需传送该图像在基图像上的权重即可恢复原图像。不同类型的基图像中的第一个一般具有一致性。其余的则依赖于构造它们的标准正交函数集。这些基图像可以用于表示大量图像, 而依赖于图像本身的特征图像只适合于表达原图。此外, 在比特流上的优势来源于基图像是预先匹配的, 不需单独为每幅图像分别存储和传输。

得益于离散的标准正交函数集的特点, 上述的基结构在硬件中的实现十分方便。迄今为止, 具有以复指数函数为基的正交集特点的结构最受欢迎。用这样的函数表达图像称为离散傅里叶变换。之所以受欢迎是由于其简便性, 通过权重的处理就能控制用于恢复原图的基图像。例如, 需要省略某些细节时, 可以巧妙地对图像进行卷积操作, 达到平滑的作用。

第3章 图像的统计描述

3.1 本章概述

本章从信号处理的观点出发，提供图像统计描述的必要的背景知识。

3.2 为什么需要对图像进行统计描述

在各种应用中，经常需要大量处理某类图像；例如，X光图片、交通场景图片等。在处理同一类图片中，每一幅图片跟其他的图片可能都是不同的，但在同一时间里同类图片可能拥有某些相同的特征。因此，我们需要对图像进行统计描述，来捕捉这些共同特征并且应用这些特征来表示图像，使之占用更少的比特并且重构时平均误差最小。

第一个想法是如果同一幅图像或相似的图像集需要被多次传输并重构，我们应使重构图像时的均方误差尽量小，而不是单独最小化每一幅图像的平方误差。第二个想法是我们用来表示图像的数据必须是无关联的。这两个想法导致图像的统计描述的产生。

3.3 是否存在一个用不相关数据来表示的图像变换可以按最小均方误差来近似图像

答案是肯定的！这个变换我们称为KL变换(Karhunen-Loeve transform)或霍特林变换(Hotelling transform)。这种方法源于将图像看做随机场的实例。

89

3.4 什么是随机场

随机场是一个定义在空间上的函数，这个函数在每个空间点都分配了一个随机变量。

3.5 什么是随机变量

随机变量(random variable)是分配给一个随机实验结果的值。

3.6 如何描述随机变量

随机变量是以它们的分布函数(distribution function)的形式描述的，而分布函数是以事件发生的概率来定义的。一个事件是随机实验结果的一个集合。

3.7 什么是事件的概率

一个事件发生的概率(probability)是一个非负的数字，它有下列的性质：

- 包括随机实验所有可能输出结果的事件发生概率为1。
- 两个没有交集的事件发生的概率等于这两个事件分别发生的概率之和。

3.8 什么是随机变量的分布函数

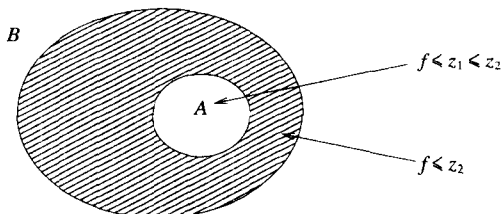
一个随机变量 f 的分布函数是一个函数。这个函数给出随机变量 f 小于函数变量值的可能性有多大:

$$\underbrace{P_f(z)}_{\text{随机变量 } f \text{ 的分布函数}} = \underbrace{\mathcal{P}}_{\text{概率}} \{ \underbrace{f}_{\text{随机变量}} < \underbrace{z}_{\text{数值}} \} \quad (3-1)$$

明显地, $P_f(-\infty) = 0$, $P_f(+\infty) = 1$ 。

例3.1 若 $Z_1 < Z_2$, 证明 $P_f(Z_1) \leq P_f(Z_2)$ 。

假设 A 是 $f < Z_1$ 的事件 (也就是输出结果的集合), B 是 $f < Z_2$ 的事件。由于 $Z_1 < Z_2$, $A \subseteq B$
 90 $\Rightarrow B = (B - A) \cup A$; 也就是事件 $(B - A)$ 和 A 没有交集 (见下一页的图)



那么通过事件概率定义中的性质(B)得

$$\begin{aligned} \mathcal{P}(B) &= \mathcal{P}(B - A) + \mathcal{P}(A) \Rightarrow \\ P_f(z_2) &= \underbrace{\mathcal{P}(B - A)}_{\text{非负数}} + P_f(z_1) \Rightarrow \end{aligned}$$

$$P_f(z_2) \geq P_f(z_1) \quad (3-2)$$

例3.2 证明

$$\mathcal{P}(z_1 < f < z_2) = P_f(z_2) - P_f(z_1)$$

根据例3.1中的符号, 当随机实验的结果属于 $B - A$ 时有 $Z_1 < f < Z_2$ (上图的阴影部分); 也就是 $P(Z_1 < f < Z_2) = P_f(B - A)$ 。由 $B = (B - A) \cup A$, $P_f(B - A) = P_f(B) - P_f(A)$ 可证明结论成立。

3.9 什么是随机变量取特定值的概率

如果随机变量从实数集里面取值, 它取特定值的概率是0。(在例3.2的结果中若设 $f = Z_1$

$=Z_2$ 可得)但是,它也可能在无穷小范围内取到非零值。这个可以由它的概率密度函数(probability density function)来表示。

91

3.10 什么是随机变量的概率密度函数

一个随机变量的分布函数的导数称为随机变量的概率密度函数

$$p_f(z) \equiv \frac{dP_f(z)}{dz} \quad (3-3)$$

随机变量 f 的期望值或均值(expected or mean value)定义如下:

$$\mu_f \equiv E\{f\} \equiv \int_{-\infty}^{\infty} z p_f(z) dz \quad (3-4)$$

方差(variance)定义如下:

$$\sigma_f^2 \equiv E\{(f - \mu_f)^2\} \equiv \int_{-\infty}^{+\infty} (z - \mu_f)^2 p_f(z) dz \quad (3-5)$$

标准差(standard deviation)是方差的正平方根,也就是 σ_f 。

3.11 如何描述多个随机变量

如果有 n 个随机变量,我们可以定义它们的联合分布函数(joint distribution function):

$$P_{f_1 f_2 \dots f_n}(z_1, z_2, \dots, z_n) \equiv \mathcal{P}\{f_1 < z_1, f_2 < z_2, \dots, f_n < z_n\} \quad (3-6)$$

同样可以定义它们的联合概率密度函数(joint probability density function):

$$p_{f_1 f_2 \dots f_n}(z_1, z_2, \dots, z_n) \equiv \frac{\partial^n P_{f_1 f_2 \dots f_n}(z_1, z_2, \dots, z_n)}{\partial z_1 \partial z_2 \dots \partial z_n} \quad (3-7)$$

3.12 n 个随机变量之间的可能关系是怎样的

如果 n 个随机变量的分布函数可以写成:

$$P_{f_1 f_2 \dots f_n}(z_1, z_2, \dots, z_n) = P_{f_1}(z_1) P_{f_2}(z_2) \dots P_{f_n}(z_n) \quad (3-8)$$

那么,这些随机变量称为独立的(independent)。它们被称为不相关的(uncorrelated),如果满足:

$$E\{f_i f_j\} = E\{f_i\} E\{f_j\}, \forall i, j, i \neq j \quad (3-9)$$

任意两个随机变量是正交的(orthogonal),如果满足:

$$E\{f_i f_j\} = 0 \quad (3-10)$$

任意两个随机变量的协方差(covariance)可定义为:

$$c_{ij} \equiv E\{(f_i - \mu_{f_i})(f_j - \mu_{f_j})\} \quad (3-11)$$

92

• 例3.3 证明若两随机变量的协方差 c_{ij} 为零, 这两个变量就是无关联的。

展开协方差定义中公式的右边, 可得到:

$$\begin{aligned}
 c_{ij} &= E\{f_i f_j - \mu_{f_i} f_j - \mu_{f_j} f_i + \mu_{f_i} \mu_{f_j}\} \\
 &= E\{f_i f_j\} - \mu_{f_i} E\{f_j\} - \mu_{f_j} E\{f_i\} + \mu_{f_i} \mu_{f_j} \\
 &= E\{f_i f_j\} - \mu_{f_i} \mu_{f_j} - \mu_{f_j} \mu_{f_i} + \mu_{f_i} \mu_{f_j} \\
 &= E\{f_i f_j\} - \mu_{f_i} \mu_{f_j}
 \end{aligned} \tag{3-12}$$

注意, 对一个固定数取期望的时候, 不会对它产生影响。也就是 $E\{\mu_{f_i}\} = \mu_{f_i}$ 。若 $c_{ij} = 0$, 可得到:

$$E\{f_i f_j\} = \mu_{f_i} \mu_{f_j} = E\{f_i\} E\{f_j\} \tag{3-13}$$

上式证明, f_i 和 f_j 是不相关的。

3.13 如何定义随机场

如果在一个二维空间的每个点都定义一个随机变量, 则得到一个二维的随机场(random field)。定义随机变量时所处的空间位置是随机场的一个参数:

$$f(\mathbf{r}; \omega_i) \tag{3-14}$$

对于固定值 \mathbf{r} , 这个函数就是一个随机变量, 但对于固定值 ω_i (输出结果) 就是平面的一个二维函数, 例如一个图像。当 ω_i 遍历统计实验中的所有可能结果的时候, 随机场表示一系列的图像。另一方面, 对一个给定的输出结果, (固定 ω_i), 随机场给出一幅图片中不同位置的灰度值。

93 • 例3.4 用骰子做一组实验。在每个实验中, 扔四次骰子。十六个实验的输出结果 $\{\omega_1, \omega_2, \omega_3, \omega_4\}$ 给定如下:

$$\begin{aligned}
 &\{1, 2, 1, 6\}, \{3, 5, 2, 4\}, \{3, 4, 6, 6\}, \{1, 1, 3, 2\}, \{3, 4, 4, 4\}, \{2, 6, 4, 2\}, \\
 &\{1, 5, 3, 6\}, \{1, 2, 6, 4\}, \{6, 5, 2, 4\}, \{3, 2, 5, 6\}, \{1, 2, 4, 5\}, \{5, 1, 1, 6\}, \\
 &\{2, 5, 3, 1\}, \{3, 1, 5, 6\}, \{1, 2, 1, 5\}, \{3, 2, 5, 4\}
 \end{aligned}$$

若 \mathbf{r} 是一个取下面数值的二维向量

$$\begin{aligned}
 &\{(1, 1), (1, 2), (1, 3), (1, 4), (2, 1), (2, 2), (2, 3), (2, 4), \\
 &(3, 1), (3, 2), (3, 3), (3, 4), (4, 1), (4, 2), (4, 3), (4, 4)\}
 \end{aligned}$$

给定由随机场 $f(\mathbf{r}; \omega_i)$ 定义的一系列图像。

第一幅图像是由每次实验在对应位置的第一个输出结果组成的, 第二幅图像是由每次实验在对应位置的第二个输出结果组成, 其他图像依此类推。得到的全部图像如下:

$$\begin{pmatrix} 1 & 3 & 3 & 1 \\ 3 & 2 & 1 & 1 \\ 6 & 3 & 1 & 5 \\ 2 & 3 & 1 & 3 \end{pmatrix} \begin{pmatrix} 2 & 5 & 4 & 1 \\ 4 & 6 & 5 & 2 \\ 5 & 2 & 2 & 1 \\ 5 & 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} 1 & 2 & 6 & 3 \\ 4 & 4 & 3 & 6 \\ 2 & 5 & 4 & 1 \\ 3 & 5 & 1 & 5 \end{pmatrix} \begin{pmatrix} 6 & 4 & 6 & 2 \\ 4 & 2 & 6 & 4 \\ 4 & 6 & 5 & 6 \\ 1 & 6 & 5 & 4 \end{pmatrix} \quad (3-15)$$

3.14 如何联系同一随机场中两个随机变量

如果固定 \mathbf{r} , 那么一个随机场就变成一个随机变量。它的期望值只依赖于 \mathbf{r} :

$$\mu_f(\mathbf{r}) = E\{f(\mathbf{r}; \omega_i)\} = \int_{-\infty}^{+\infty} z p_f(z; \mathbf{r}) dz \quad (3-16)$$

因为不同的 \mathbf{r} 值有不同的随机变量, $f(\mathbf{r}_1; \omega_i)$ 和 $f(\mathbf{r}_2; \omega_i)$, 定义相关性为自相关(autocorrelation) (用“自”是因为两个变量都来自相同的随机场):

$$R_{ff}(\mathbf{r}_1, \mathbf{r}_2) = E\{f(\mathbf{r}_1; \omega_i) f(\mathbf{r}_2; \omega_i)\} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} z_1 z_2 p_f(z_1, z_2; \mathbf{r}_1, \mathbf{r}_2) dz_1 dz_2 \quad (3-17)$$

自协方差(autocovariance) $C(\mathbf{r}_1, \mathbf{r}_2)$ 定义如下:

$$C_{ff}(\mathbf{r}_1, \mathbf{r}_2) = E\{[f(\mathbf{r}_1; \omega_i) - \mu_f(\mathbf{r}_1)][f(\mathbf{r}_2; \omega_i) - \mu_f(\mathbf{r}_2)]\} \quad (3-18) \quad \boxed{94}$$

• 例3.5 证明:

$$C_{ff}(\mathbf{r}_1, \mathbf{r}_2) = R_{ff}(\mathbf{r}_1, \mathbf{r}_2) - \mu_f(\mathbf{r}_1)\mu_f(\mathbf{r}_2)$$

从式(3-18)开始:

$$\begin{aligned} C_{ff}(\mathbf{r}_1, \mathbf{r}_2) &= E\{[f(\mathbf{r}_1; \omega_i) - \mu_f(\mathbf{r}_1)][f(\mathbf{r}_2; \omega_i) - \mu_f(\mathbf{r}_2)]\} \\ &= E\{f(\mathbf{r}_1; \omega_i)f(\mathbf{r}_2; \omega_i) - f(\mathbf{r}_1; \omega_i)\mu_f(\mathbf{r}_2) - \mu_f(\mathbf{r}_1)f(\mathbf{r}_2; \omega_i) \\ &\quad + \mu_f(\mathbf{r}_1)\mu_f(\mathbf{r}_2)\} \\ &= E\{f(\mathbf{r}_1; \omega_i)f(\mathbf{r}_2; \omega_i)\} - E\{f(\mathbf{r}_1; \omega_i)\}\mu_f(\mathbf{r}_2) - \mu_f(\mathbf{r}_1)E\{f(\mathbf{r}_2; \omega_i)\} \\ &\quad + \mu_f(\mathbf{r}_1)\mu_f(\mathbf{r}_2) \\ &= R_{ff}(\mathbf{r}_1, \mathbf{r}_2) - \mu_f(\mathbf{r}_1)\mu_f(\mathbf{r}_2) - \mu_f(\mathbf{r}_1)\mu_f(\mathbf{r}_2) + \mu_f(\mathbf{r}_1)\mu_f(\mathbf{r}_2) \\ &= R_{ff}(\mathbf{r}_1, \mathbf{r}_2) - \mu_f(\mathbf{r}_1)\mu_f(\mathbf{r}_2) \end{aligned} \quad (3-19)$$

3.15 如何联系两个随机场中两个随机变量

如果我们有二个随机场, 例如, 由二个不同的随机实验产生的两组不同的图像, 用 f 和 g 来表示。我们可以定义它们的互相关性(cross correlation)为

$$R_{fg}(\mathbf{r}_1, \mathbf{r}_2) = E\{f(\mathbf{r}_1; \omega_i)g(\mathbf{r}_2; \omega_j)\} \quad (3-20)$$

它们的互协方差(cross covariance)为

$$\begin{aligned} C_{fg}(\mathbf{r}_1, \mathbf{r}_2) &= E\{[f(\mathbf{r}_1; \omega_i) - \mu_f(\mathbf{r}_1)][g(\mathbf{r}_2; \omega_j) - \mu_g(\mathbf{r}_2)]\} \\ &= R_{fg}(\mathbf{r}_1, \mathbf{r}_2) - \mu_f(\mathbf{r}_1)\mu_g(\mathbf{r}_2) \end{aligned} \quad (3-21)$$

两个随机场称为不相关的, 如果对于任意的 $\mathbf{r}_1, \mathbf{r}_2$ 有

$$C_{fg}(\mathbf{r}_1, \mathbf{r}_2) = 0 \quad (3-22)$$

这等价于

$$E\{f(\mathbf{r}_1; \omega_i)g(\mathbf{r}_2; \omega_j)\} = E\{f(\mathbf{r}_1; \omega_i)\}E\{g(\mathbf{r}_2; \omega_j)\} \quad (3-23)$$

• 例3.6 证明对于两个不相关的随机场有

$$E\{f(\mathbf{r}_1; \omega_i)g(\mathbf{r}_2; \omega_j)\} = E\{f(\mathbf{r}_1; \omega_i)\}E\{g(\mathbf{r}_2; \omega_j)\}$$

一般可以从不相关随机场的定义($C_{fg}(\mathbf{r}_1, \mathbf{r}_2) = 0$)和表达式:

$$C_{fg}(\mathbf{r}_1, \mathbf{r}_2) = E\{f(\mathbf{r}_1; \omega_i)g(\mathbf{r}_2; \omega_j)\} - \mu_f(\mathbf{r}_1)\mu_g(\mathbf{r}_2) \quad (3-24)$$

用例3.5中相似的方法证得上式。

3.16 既然我们总是只有图像的一个版本, 如何计算以前定义中的期望值

假设图像是齐次(homogeneous)随机场, 并且是遍历的(ergodic)。遍历性定理允许我们用一幅图像的空间统计来代替全体的统计。

3.17 什么时候随机场是齐次的

如果随机场的期望不依赖于 \mathbf{r} , 并且它的自相关函数是平移不变的, 那么这个随机场就称为齐次的。

平移不变自相关函数只依赖于一个变量, 即位置的相对移动。随机场的值可由下式计算:

$$R_{ff}(\mathbf{r}_0) = E\{f(\mathbf{r}; \omega_i)f(\mathbf{r} + \mathbf{r}_0; \omega_i)\} \quad (3-25)$$

• 例3.7 证明一个齐次随机场的自相关函数 $R(\mathbf{r}_1, \mathbf{r}_2)$ 只依赖于向量差 $\mathbf{r}_1 - \mathbf{r}_2$

齐次随机场的自相关函数是平移不变的。所以, 对于任意的平移向量 \mathbf{r}_0 我们可写成

$$\begin{aligned} R_{ff}(\mathbf{r}_1, \mathbf{r}_2) &= E\{f(\mathbf{r}_1; \omega_i)f(\mathbf{r}_2; \omega_i)\} = E\{f(\mathbf{r}_1 + \mathbf{r}_0; \omega_i)f(\mathbf{r}_2 + \mathbf{r}_0; \omega_i)\} \\ &= R_{ff}(\mathbf{r}_1 + \mathbf{r}_0, \mathbf{r}_2 + \mathbf{r}_0) \quad \forall \mathbf{r}_0 \end{aligned} \quad (3-26)$$

选择 $\mathbf{r}_0 = -\mathbf{r}_2$, 对于一个齐次随机场, 我们可得到

$$R_{ff}(\mathbf{r}_1, \mathbf{r}_2) = R_{ff}(\mathbf{r}_1 - \mathbf{r}_2, 0) = R_{ff}(\mathbf{r}_1 - \mathbf{r}_2) \quad (3-27)$$

3.18 如何计算随机场的空间统计量

给定一个随机场，我们可以定义它的空间平均值为

$$\mu(\omega_i) \equiv \lim_{S \rightarrow \infty} \frac{1}{S} \int_S f(\mathbf{r}; \omega_i) dxdy \quad (3-28)$$

在这里 \int_S 是在整个空间 S 上的积分，并且 $\mathbf{r} = (x, y)$ 。很明显， $\mu(\omega_i)$ 的值是 f 输出结果的函数。也就是说， $\mu(\omega_i)$ 是一个随机变量。

随机场的空间自相关函数定义为

$$R(\mathbf{r}_0; \omega_i) \equiv \lim_{S \rightarrow \infty} \frac{1}{S} \int_S f(\mathbf{r}; \omega_i) f(\mathbf{r} + \mathbf{r}_0; \omega_i) dxdy \quad (3-29)$$

这也是随机变量。

3.19 什么时候随机场是遍历性的

当随机场对于均值和自相关函数是遍历性的，则随机场是遍历性的。

3.20 什么时候随机场对于均值是遍历性的

称一个随机场对于均值是遍历性的，如果它是齐次并且它由式(3-28)定义的空间均值是独立于函数输出的。也就是说，它是一个常数，并且这个常数等于式(3-16)所定义的全体平均。

$$E\{f(\mathbf{r}; \omega_i)\} = \lim_{S \rightarrow \infty} \frac{1}{S} \int_S f(\mathbf{r}; \omega_i) dxdy = \mu = \text{常数} \quad (3-30)$$

3.21 什么时候随机场对于自相关函数是遍历性的

称随机场对于自相关函数是遍历性的，当它是齐次的并且由式(3-29)所定义的空间自相关函数，是独立于在 f 上所做的实验的结果，只与位移 \mathbf{r}_0 有关，并且等于式(3-25)所定义的自相关函数。

97

$$E\{f(\mathbf{r}; \omega_i) f(\mathbf{r} + \mathbf{r}_0; \omega_i)\} = \lim_{S \rightarrow \infty} \frac{1}{S} \int_S f(\mathbf{r}; \omega_i) f(\mathbf{r} + \mathbf{r}_0; \omega_i) dxdy = R(\mathbf{r}_0) \quad (3-31)$$

• 例3.8 假设存在遍历性，计算下面图像的自相关矩阵：

$$\begin{pmatrix} 1 & 2 & 1 \\ 1 & 2 & 1 \\ 1 & 2 & 1 \end{pmatrix}$$

3 × 3 图像有下面的格式：

$$\begin{pmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{pmatrix} \quad (3-32)$$

为了计算它的自相关函数, 我们将它按列堆成一个列向量:

$$\mathbf{g} = (g_{11} \ g_{21} \ g_{31} \ g_{12} \ g_{22} \ g_{32} \ g_{13} \ g_{23} \ g_{33})^T \quad (3-33)$$

自相关矩阵定义如下: $C = E\{\mathbf{g}\mathbf{g}^T\}$ 。由于事先假定了遍历性, 所以我们只需要计算在图像相同位置上像素对的平均, 而不需要对所有可能图像进行平均。自相关矩阵有如下的结构:

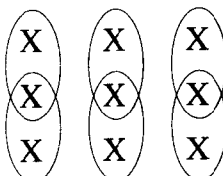
	g_{11}	g_{21}	g_{31}	g_{12}	g_{22}	g_{32}	g_{13}	g_{23}	g_{33}
g_{11}	A	B	C	D	E	F	G	H	I
g_{21}	B	A	B	J	D	E	K	G	H
g_{31}	C	B	A	L	J	D	M	K	G
g_{12}	D	J	L	A	B	C	D	E	F
g_{22}	E	D	J	B	A	B	J	D	E
g_{32}	F	E	D	C	B	A	L	J	D
g_{13}	G	K	M	D	J	L	A	B	C
g_{23}	H	G	K	E	D	J	B	A	B
g_{33}	I	H	G	F	E	D	C	B	A

(3-34)

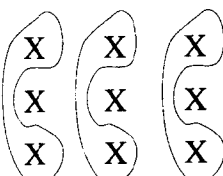
矩阵第一行和第一列显示了图像中元素间的关联, 而得到矩阵中对应的项。A是元素平方的均值, 计算可得

$$\frac{\sum_{ij} g_{ij}^2}{9} = \frac{6 \times 1^2 + 3 \times 2^2}{9} = \frac{18}{9} = 2 \quad (3-35)$$

B是垂直邻点积的均值。有6对这样的点。我们必须对这些积的值进行求和并平均。问题在于, 我们应该除以实际的垂直邻点对数, 也就是6, 还是除以总的像素数, 也就是9。这个问题是会影响到矩阵中除了A以外的所有项的计算。如果我们除以真正的点对的数目, 大部分比较远的邻点的关系将被夸大。所以, 我们选择除以图像中总的像素数, 这会使较远的邻点的关系变弱, 但这是非常重要的。这个问题的出现是因为图像的大小有限。注意到式(3-29)和式(3-28)真正应用是在无限尺寸的图像上的。这个问题在处理一个很小图像时将变得更加重要。因为它的边缘影响被夸大了。

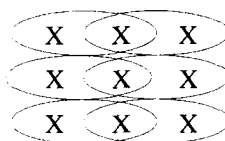
$$B = \frac{4 \times 1 + 2 \times 4}{9} = \frac{12}{9} = 1.33$$


C是垂直方向上间隔点的积的均值。共有3对:

$$C = \frac{2 \times 1 + 4}{9} = \frac{6}{9} = 0.67$$


D是水平方向邻点的积的均值。共有6对:

$$D = \frac{6 \times 2}{9} = 1.33$$



E 是对角方向邻点的积的均值，共有4对

99

$E:$

$E = \frac{4 \times 2}{9} = 0.89$

$F:$

$F = \frac{2 \times 2}{9} = 0.44$

$G:$

$G = \frac{3 \times 1}{9} = 0.33$

$H:$

$H = \frac{2 \times 1}{9} = 0.22$

100

$I:$

$I = \frac{1 \times 1}{9} = 0.11$

$J:$

$J = \frac{4 \times 2}{9} = 0.89$

$K:$

$K = \frac{2 \times 1}{9} = 0.22$

$$L: \begin{array}{ccc} \text{X} & \text{X} & \text{X} \\ \text{X} & \text{X} & \text{X} \\ \text{X} & \text{X} & \text{X} \end{array} \quad L = \frac{2 \times 2}{9} = 0.44$$

$$M: \begin{array}{ccc} & \text{X} & \text{X} & \text{X} \\ \text{X} & & \text{X} & \\ \text{X} & & \text{X} & \end{array} \quad M = \frac{1}{9} = 0.11$$

所以, 自相关矩阵是

$$\begin{pmatrix} 2 & 1.33 & 0.67 & 1.33 & 0.89 & 0.44 & 0.33 & 0.22 & 0.11 \\ 1.33 & 2 & 1.33 & 0.89 & 1.33 & 0.89 & 0.22 & 0.33 & 0.22 \\ 0.67 & 1.33 & 2 & 0.44 & 0.89 & 1.33 & 0.11 & 0.22 & 0.22 \\ 1.33 & 0.89 & 0.44 & 2 & 1.33 & 0.67 & 1.33 & 0.89 & 0.44 \\ 0.89 & 1.33 & 0.89 & 1.33 & 2 & 1.33 & 0.89 & 1.33 & 0.89 \\ 0.44 & 0.89 & 1.33 & 0.67 & 1.33 & 2 & 0.44 & 0.89 & 1.33 \\ 0.33 & 0.22 & 0.11 & 1.33 & 0.89 & 0.44 & 2 & 1.33 & 0.67 \\ 0.22 & 0.33 & 0.22 & 0.89 & 1.33 & 0.89 & 1.33 & 2 & 1.33 \\ 0.11 & 0.22 & 0.33 & 0.44 & 0.89 & 1.33 & 0.67 & 1.33 & 2 \end{pmatrix}$$

例3.9 给定下列一组图像:

$$\begin{pmatrix} 5 & 4 & 6 & 2 \\ 5 & 3 & 4 & 3 \\ 6 & 6 & 7 & 1 \\ 5 & 4 & 2 & 3 \end{pmatrix}, \begin{pmatrix} 4 & 2 & 2 & 1 \\ 7 & 2 & 4 & 9 \\ 3 & 5 & 4 & 5 \\ 4 & 6 & 6 & 2 \end{pmatrix}, \begin{pmatrix} 3 & 5 & 2 & 3 \\ 5 & 4 & 4 & 3 \\ 2 & 2 & 6 & 6 \\ 6 & 5 & 4 & 6 \end{pmatrix}, \begin{pmatrix} 6 & 4 & 2 & 8 \\ 3 & 5 & 6 & 4 \\ 4 & 3 & 2 & 2 \\ 5 & 3 & 3 & 4 \end{pmatrix},$$

$$\begin{pmatrix} 4 & 3 & 5 & 4 \\ 6 & 5 & 6 & 2 \\ 4 & 3 & 3 & 4 \\ 3 & 3 & 6 & 5 \end{pmatrix}, \begin{pmatrix} 4 & 5 & 4 & 5 \\ 1 & 6 & 2 & 6 \\ 4 & 8 & 4 & 4 \\ 1 & 3 & 2 & 7 \end{pmatrix}, \begin{pmatrix} 2 & 7 & 6 & 4 \\ 2 & 4 & 2 & 4 \\ 6 & 3 & 4 & 7 \\ 4 & 3 & 6 & 2 \end{pmatrix}, \begin{pmatrix} 5 & 3 & 6 & 6 \\ 4 & 4 & 5 & 2 \\ 4 & 2 & 3 & 4 \\ 5 & 5 & 4 & 4 \end{pmatrix}$$

这组图像对于均值是遍历性的吗? 它对于自相关函数遍历性吗?

101

这组图像对于均值是遍历性的。因为每幅图像的均值是4.125, 而在所有8幅图像上的同一位置的均值也是4.125。它对于自相关函数不是遍历性的。为了证明这一点, 我们先计算自相关矩阵上的每一个元素。比如说元素 $E\{g_{23}g_{34}\}$ 。这个元素表示的是在所有图像的位置(2, 3)和(3, 4)上值的内积的均值:

$$\begin{aligned} E\{g_{23}g_{34}\} &= \frac{4 \times 1 + 4 \times 5 + 4 \times 6 + 6 \times 2 + 6 \times 4 + 2 \times 4 + 2 \times 7 + 5 \times 4}{8} \\ &= \frac{4 + 20 + 24 + 12 + 24 + 8 + 14 + 20}{8} = \frac{126}{8} = 15.75 \end{aligned}$$

这应该等于自相关函数上表达点对的空间均值的元素, 这些点对在左上角到右下角的对角方向上相邻。考虑图像集中的最后一幅图像。我们有

$$\begin{aligned}
 & \langle g_{ij}g_{i+1,j+1} \rangle \\
 &= \frac{5 \times 4 + 3 \times 5 + 6 \times 2 + 4 \times 2 + 4 \times 3 + 5 \times 4 + 4 \times 5 + 2 \times 4 + 3 \times 4}{16} \\
 &= \frac{20 + 15 + 12 + 8 + 12 + 20 + 20 + 8 + 12}{16} = 7.9375
 \end{aligned}$$

这两个值是不同的，所以图像集对于自相关函数不是遍历的。

•
•
•

3.22 遍历性的含义是什么

如果一个图像集是遍历性的，则可以通过计算这个图像集中的任意一幅图像的空间均值和自相关函数，得到图像集的均值和自相关函数。

例如，假设我们有 M 幅相似类型图像的集合 $\{g_1(x, y), g_2(x, y), \dots, g_M(x, y)\}$ ，这些图像的均值和自相关函数可以通过对所有的图像取平均得到。另一方面，如果假定存在遍历性的话，我们可以随便取出一幅图像，通过空间平均得到均值和自相关函数。如果不同图像的变异性在统计意义上跟单幅图像一样的话，这个结果是正确的。

3.23 如何利用遍历性来减少表示一幅图像所需要的比特数

假设有一个遍历性的图像 g ，我们打算将它通过一个通信通道发送出去。希望发送的图像的各个比特之间都是无关联的，这样就不用重复发送已经发送的信息。也就是说，给定传送的比特的个数，我们希望最大化传送的图像信息。

102

一个有着这种性质的随机场的自相关函数有着特殊的形式。在我们决定如何对图像进行变换，以使它是由不相关的点值组成之后。通过遍历性我们对一幅单独的图像统计就可以得到必要的变换，而不需要对整个图像集进行统计。

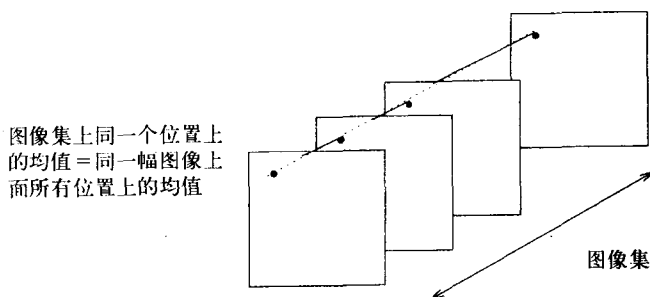


图3-1 坚果壳似的遍历性

3.24 具有不相关随机变量的随机场的自相关函数的形式是怎样的

两个定义在位置 \mathbf{r}_1 和 \mathbf{r}_2 的随机变量的自相关函数 $R_{ff}(\mathbf{r}_1, \mathbf{r}_2)$ 等于 $E\{f(\mathbf{r}_1; \omega_i)\}E\{f(\mathbf{r}_2; \omega_i)\}$ ，如果这两个随机变量是无关联的（见例3.3）。如果我们只处理均值为0的随机变量（也就是

$E\{f(\mathbf{r}_1, \omega_i)\} = E\{f(\mathbf{r}_2, \omega_i)\} = 0$), 那么对变量所有的值自相关矩阵将为0, 当 $\mathbf{r}_1 = \mathbf{r}_2$ 时例外。当 $\mathbf{r}_1 = \mathbf{r}_2$ 时, 自相关矩阵等于 $E\{f(\mathbf{r}_1, \omega_i)^2\}$, 即, 随机变量在位置 \mathbf{r}_1 处的方差。

如果一个图像 \mathbf{g} 是由一个列向量所表示的, 我们将不用向量 $\mathbf{r}_1, \mathbf{r}_2$ 来标明像素的位置, 而只需要用整数标号 i, j 来标明每一个列向量元素。自相关函数 R_{gg} 变成一个二维矩阵, 对于不相关0均值的随机变量来说, 这个矩阵将是对角阵, 而且它的对角上面的非0元素等于在每一个像素点上的方差。(如在例3.8自相关矩阵中, $A \neq 0$ 但所有其他元素都为0)

3.25 如何变换图像使得它的自相关矩阵是对角的

103

假设初始图像为 \mathbf{g} 而变换后的图像为 $\tilde{\mathbf{g}}$ 。我们将利用 \mathbf{g} 和 $\tilde{\mathbf{g}}$ 对应的向量形式。也就是说, 将两个矩阵堆积起来, 将其中一个的列放在另一个的相对列的上面, 首尾相连起来构成两个 $N^2 \times 1$ 的向量。假定所需的变换有形式:

$$\tilde{\mathbf{g}} = A(\mathbf{g} - \mathbf{m}) \quad (3-36)$$

这里, 变换矩阵 A 是 $N^2 \times N^2$ 的, 而任意向量 \mathbf{m} 是 $N^2 \times 1$ 的。由于假定图像是遍历性的。变换图像的均值向量给定如下:

$$\mu_{\tilde{\mathbf{g}}} = E\{\tilde{\mathbf{g}}\} = E\{A(\mathbf{g} - \mathbf{m})\} = AE\{\mathbf{g}\} - A\mathbf{m} = A(\mu_{\mathbf{g}} - \mathbf{m}) \quad (3-37)$$

其中注意到事实: 如果 \mathbf{m} 是非随机向量, 那么期望值算子对它是没有影响的。注意我们用同样的记号表示期望值和集的平均。由于假定了遍历性, $E\{\tilde{\mathbf{g}}\}$ 就是找到图像 $\tilde{\mathbf{g}}$ 的平均灰度值, 是一个由图像平均灰度值组成的 $N^2 \times 1$ 的向量。如果事先没有假定遍历性, 那么 $E\{\tilde{\mathbf{g}}\}$ 就是对所有的图像 $\tilde{\mathbf{g}}$ 的平均。

在式(3-37)中, 我们很方便地可以选择 $\mathbf{m} = \mu_{\mathbf{g}} = E\{\mathbf{g}\}$, 那么 $\mu_{\tilde{\mathbf{g}}} = 0$ 。也就是, 变换后的图像将具有0均值。

图像 $\tilde{\mathbf{g}}$ 的自相关函数将与它的协方差函数一样。并定义如下:

$$\begin{aligned} C_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}} &= E\{\tilde{\mathbf{g}}\tilde{\mathbf{g}}^T\} = E\{A(\mathbf{g} - \mu_{\mathbf{g}})[A(\mathbf{g} - \mu_{\mathbf{g}})]^T\} \\ &= E\{A(\mathbf{g} - \mu_{\mathbf{g}})(\mathbf{g} - \mu_{\mathbf{g}})^T A^T\} \\ &= A \underbrace{E\{(\mathbf{g} - \mu_{\mathbf{g}})(\mathbf{g} - \mu_{\mathbf{g}})^T\}}_{\text{变换前的图像的协方差}} A^T \end{aligned} \quad (3-38)$$

再一次注意, 期望值算子是指空间平均, 并且因为矩阵 A 不是随机场, 所以不受期望值算子的影响。

所以, $C_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}} = AC_{\mathbf{g}\mathbf{g}}A^T$ 。明显可以看出 $C_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}}$ 是未变换的图像的协方差矩阵的对角化形式。如果转换矩阵 A 每一行都是由图像的协方差矩阵的特征向量形成的, 则很容易就可以实现这种对角化并且 $C_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}}$ 的对角元素是同一矩阵的特征值。由于我们事先假定了遍历性, 则图像的协方差矩阵就可以从图像自身求得(不需要大量相似的图像)。

3.26 遍历性的假设是现实的吗

遍历性的假设是不现实的。期望一个单独的图像大到足以包括其内容中出现的所有的变

动, 并且捕捉到图像集所表现的所有多样性是很不现实的。只有由纯随机噪声组成的图像可以满足这个假设。所以, 人们经常将图像分成很多相同的小块, 去除这些小块的噪声影响, 它们是均匀的, 因此适用遍历性的假设。

104

B3.1 我们如何计算一个图像的空间自相关矩阵?

为了定义一个图像的空间自相关矩阵, 我们必须先建立图像像素位置与图像的向量位置表示的对应关系。由于一个图像的向量表示是通过将图像各列堆叠得到的, 在位置 (k_i, l_i) 的像素将是堆叠后的向量的第 i 个元素, 其中:

$$i = (l_i - 1)N + k_i + 1 \quad (3-39)$$

这里 N 是图像中每一列元素的数量。由上式可得到可用 i 表示 k_i 和 l_i :

$$\begin{aligned} k_i &= (i - 1) \text{ modulo } N \\ l_i &= 1 + \frac{i - 1 - k_i}{N} \end{aligned} \quad (3-40)$$

自相关矩阵中的元素 C_{ij} 可以写成

$$C_{ij} \equiv \langle g_i g_j \rangle \quad (3-41)$$

其中 $\langle \rangle$ 意味着在相同关联位置上点对的平均, 也就是说 $l_i - l_j$ 和 $k_i - k_j$ 是一样的。首先, 我们计算 $l_i - l_j$ 和 $k_i - k_j$:

$$\begin{aligned} k_0 &\equiv k_i - k_j = (i - 1) \text{ modulo } N - (j - 1) \text{ modulo } N \\ l_0 &\equiv l_i - l_j = \frac{i - j - k_0}{N} \end{aligned} \quad (3-42)$$

所以,

$$C_{ij} = \langle g_{kl} g_{k-k_0, l-l_0} \rangle \quad (3-43)$$

$k, l, k-k_0$ 和 $l-l_0$ 取的值必须在允许的范围内, 对于一幅 $N \times N$ 的图像来说, 也就是在范围 $[1, N]$ 内取值:

$$\begin{aligned} 1 \leq k - k_0 \leq N &\Rightarrow 1 + k_0 \leq k \leq N + k_0 \\ \text{Also } 1 \leq k \leq N &\Rightarrow \max(1, 1 + k_0) \leq k \leq \min(N, N + k_0) \end{aligned} \quad (3-44)$$

105

相似地

$$\max(1, 1 + l_0) \leq l \leq \min(N, N + l_0) \quad (3-45)$$

所以, 最后得到

$$C_{ij} = \frac{1}{N^2} \sum_{k=\max(1, 1+k_0)}^{\min(N, N+k_0)} \sum_{l=\max(1, 1+l_0)}^{\min(N, N+l_0)} g_{kl} g_{k-k_0, l-l_0} \quad (3-46)$$

在这里 k_0, l_0 是由式(3-42)给出。

• 例3.10(B) 对于图像

$$\begin{pmatrix} 3 & 5 & 2 & 3 \\ 5 & 4 & 4 & 3 \\ 2 & 2 & 6 & 6 \\ 6 & 5 & 4 & 6 \end{pmatrix}$$

计算K-L变换，假定图像具有遍历性。

图像的均值是 $\frac{66}{16} = 4.125$ 。先将图像的各个元素都减去均值，然后使用方框B3.1中得到的

公式，计算它的自相关矩阵。结果是：

$$\begin{pmatrix} 2.11 & -0.38 & -0.38 & -0.20 & 0.31 & 0.12 & 0.04 & -0.32 \\ -0.38 & 2.11 & -0.38 & -0.38 & -0.19 & 0.31 & 0.12 & 0.04 \\ -0.38 & -0.38 & 2.11 & -0.38 & 0.02 & -0.19 & 0.31 & 0.12 \\ -0.20 & -0.38 & -0.38 & 2.11 & -0.00 & 0.02 & -0.19 & 0.31 \\ 0.31 & -0.19 & 0.02 & -0.00 & 2.11 & -0.38 & -0.38 & -0.20 \\ 0.12 & 0.31 & -0.19 & 0.02 & -0.38 & 2.11 & -0.38 & -0.38 \\ 0.04 & 0.12 & 0.31 & -0.19 & -0.38 & -0.38 & 2.11 & -0.38 \\ -0.32 & 0.04 & 0.12 & 0.31 & -0.20 & -0.38 & -0.38 & 2.11 \\ -0.32 & 0.38 & 0.35 & -0.31 & 0.31 & -0.19 & 0.02 & -0.00 \\ -0.20 & -0.32 & 0.38 & 0.35 & 0.12 & 0.31 & -0.19 & 0.02 \\ -0.05 & -0.20 & -0.32 & 0.38 & 0.04 & 0.12 & 0.31 & -0.19 \\ 0.11 & -0.05 & -0.20 & -0.32 & -0.32 & 0.04 & 0.12 & 0.31 \\ -0.01 & 0.31 & 0.02 & -0.13 & -0.32 & 0.38 & 0.35 & -0.31 \\ -0.07 & -0.01 & 0.31 & 0.02 & -0.20 & -0.32 & 0.38 & 0.35 \\ -0.03 & -0.07 & -0.01 & 0.31 & -0.05 & -0.20 & -0.32 & 0.38 \\ -0.13 & -0.03 & -0.07 & -0.01 & 0.11 & -0.05 & -0.20 & -0.32 \end{pmatrix}$$

$$\begin{pmatrix} -0.32 & -0.20 & -0.05 & 0.11 & -0.01 & -0.07 & -0.03 & -0.13 \\ 0.38 & -0.32 & -0.20 & -0.05 & 0.31 & -0.01 & -0.07 & -0.03 \\ 0.35 & 0.38 & -0.32 & -0.20 & 0.02 & 0.31 & -0.01 & -0.07 \\ -0.31 & 0.35 & 0.38 & -0.32 & -0.13 & 0.02 & 0.31 & -0.01 \\ 0.31 & 0.12 & 0.04 & -0.32 & -0.32 & -0.20 & -0.05 & 0.11 \\ -0.19 & 0.31 & 0.12 & 0.04 & 0.38 & -0.32 & -0.20 & -0.05 \\ 0.02 & -0.19 & 0.31 & 0.12 & 0.35 & 0.38 & -0.32 & -0.20 \\ -0.00 & 0.02 & -0.19 & 0.31 & -0.31 & 0.35 & 0.38 & -0.32 \\ 2.11 & -0.38 & -0.38 & -0.20 & 0.31 & 0.13 & 0.04 & -0.32 \\ -0.38 & 2.11 & -0.38 & -0.38 & -0.19 & 0.31 & 0.12 & 0.04 \\ -0.38 & -0.38 & 2.11 & -0.38 & 0.02 & -0.19 & 0.31 & 0.12 \\ -0.20 & -0.38 & -0.38 & 2.11 & -0.00 & 0.02 & -0.19 & 0.31 \\ 0.31 & -0.19 & 0.02 & -0.00 & 2.11 & -0.38 & -0.38 & -0.20 \\ 0.12 & 0.31 & -0.19 & 0.02 & -0.38 & 2.11 & -0.38 & -0.38 \\ 0.04 & 0.12 & 0.31 & -0.19 & -0.38 & -0.38 & 2.11 & -0.38 \\ -0.32 & 0.04 & 0.12 & 0.31 & -0.20 & -0.38 & -0.38 & 2.11 \end{pmatrix}$$

在计算完自相关矩阵的特征向量，并将它们按照特征值的大小降序排列之后，每个特征向量对应一行构成一个变换矩阵A，图像变换以后得

$$\begin{pmatrix} 0.00 & 0.35 & 0.02 & -0.41 & -0.18 & 0.13 & 0.31 & -0.25 \\ -0.25 & 0.01 & 0.44 & -0.12 & -0.02 & -0.32 & 0.13 & 0.32 \\ -0.33 & 0.40 & -0.17 & 0.21 & -0.22 & 0.17 & -0.14 & 0.25 \\ 0.06 & -0.03 & 0.21 & -0.10 & 0.53 & -0.02 & -0.11 & -0.38 \\ 0.13 & 0.14 & -0.22 & -0.24 & 0.31 & -0.25 & -0.38 & 0.23 \\ 0.17 & -0.15 & -0.09 & 0.05 & 0.07 & -0.47 & 0.45 & -0.03 \\ -0.63 & 0.10 & 0.16 & 0.08 & 0.01 & -0.23 & 0.06 & -0.03 \\ -0.02 & 0.41 & -0.48 & 0.21 & 0.23 & -0.01 & -0.03 & -0.03 \\ 0.02 & 0.36 & 0.08 & -0.52 & -0.14 & 0.13 & 0.09 & -0.21 \\ -0.06 & -0.26 & 0.25 & -0.15 & 0.01 & 0.44 & -0.35 & 0.15 \\ -0.06 & -0.21 & -0.13 & 0.31 & -0.24 & 0.16 & 0.06 & -0.50 \\ -0.03 & 0.18 & 0.06 & 0.07 & 0.54 & 0.27 & 0.29 & 0.10 \\ 0.55 & 0.19 & 0.20 & 0.15 & -0.17 & 0.06 & 0.02 & 0.25 \\ 0.11 & 0.21 & 0.36 & 0.39 & 0.08 & 0.23 & 0.29 & 0.11 \\ 0.04 & -0.23 & -0.34 & -0.24 & -0.14 & 0.09 & 0.32 & 0.37 \\ 0.22 & 0.29 & 0.21 & 0.12 & -0.25 & -0.36 & -0.30 & -0.16 \end{pmatrix}$$

107

$$\begin{pmatrix} 0.25 & -0.31 & -0.13 & 0.18 & 0.41 & -0.02 & -0.35 & -0.00 \\ 0.32 & 0.13 & -0.32 & -0.02 & -0.12 & 0.44 & 0.01 & -0.25 \\ 0.25 & -0.17 & 0.17 & -0.22 & 0.21 & -0.17 & 0.40 & -0.33 \\ 0.38 & 0.11 & 0.02 & -0.53 & 0.10 & -0.21 & 0.03 & -0.06 \\ 0.23 & -0.38 & -0.25 & 0.31 & -0.24 & -0.22 & 0.14 & 0.13 \\ 0.03 & -0.45 & 0.47 & -0.07 & -0.05 & 0.09 & 0.15 & -0.17 \\ 0.03 & -0.06 & 0.23 & -0.01 & -0.08 & -0.16 & -0.10 & 0.63 \\ 0.03 & 0.03 & 0.01 & -0.23 & -0.21 & 0.48 & -0.41 & 0.02 \\ -0.21 & 0.09 & 0.13 & -0.14 & -0.52 & 0.08 & 0.36 & 0.02 \\ 0.15 & -0.35 & 0.44 & 0.01 & -0.15 & 0.25 & -0.26 & -0.06 \\ 0.50 & -0.06 & -0.16 & 0.24 & -0.31 & 0.13 & 0.21 & 0.06 \\ 0.10 & 0.29 & 0.27 & 0.54 & 0.07 & 0.06 & 0.18 & -0.03 \\ 0.25 & 0.02 & 0.06 & -0.17 & 0.15 & 0.20 & 0.19 & 0.55 \\ -0.11 & -0.29 & -0.23 & -0.08 & -0.39 & -0.36 & -0.21 & -0.11 \\ 0.37 & 0.32 & 0.09 & -0.14 & -0.24 & -0.34 & -0.23 & 0.04 \\ 0.16 & 0.30 & 0.36 & 0.25 & -0.12 & -0.21 & -0.29 & -0.22 \end{pmatrix}$$

对应的特征值为:

$$\begin{matrix} 3.87 & 3.70 & 3.02 & 2.96 & 2.91 & 2.85 & 2.44 & 2.16 \\ 1.87 & 1.59 & 1.39 & 1.33 & 1.19 & 1.06 & 0.85 & 0.54 \end{matrix}$$

将图像的列堆叠起来,再减去它们的均值,得到向量 $\{g - \mu_g\}$:

$$(g - \mu_g)^T = (-1.125 \quad 0.875 \quad -2.125 \quad 1.875 \quad 0.875 \quad -0.125 \quad -2.125 \quad 0.875 \quad -2.125 \quad -0.125 \quad 1.875 \quad -0.125 \quad -1.125 \quad -1.125 \quad 1.875 \quad 1.875)$$

将这个向量与矩阵A相乘,得到图像的K-L变换矩阵如下:

$$\tilde{g} = \begin{pmatrix} -3.4353 & 1.3842 & 0.5355 & 0.0363 \\ -2.9681 & -0.1112 & -0.3182 & -0.3902 \\ 1.6777 & 2.1243 & -0.7152 & -0.7986 \\ -0.9932 & 0.9922 & 0.4275 & -0.1979 \end{pmatrix}$$

108

很容易看出, 图像的均值是不为0的。它的自相关矩阵也不是对角的。好像至今所提出的理论没有起作用。事实上, 在这个例子中, 我们计算的自相关矩阵也有些问题的。

B3.2 为什么例3.10(B)中的图像违背了相关的理论呢?

K-L变换理论是建立在假设图像具有遍历性的基础上的, 所以我们能用空间的统计性质代替整体的统计性质。这是一个相当强的假设, 例子3.10(B)中的图像并没有满足这个假设。结果, 空间上的均值不等于整体的均值, 并且空间自相关矩阵也不等于整体的自相关矩阵。所以, 虽然我们很小心地让原始图像的空间均值为0, 并且对角化它的空间自相关矩阵, 但是变换后的图像并没有体现出这些特征。

为了解决这些问题, 我们尝试利用式(3-37)、式(3-36)来计算变换后的图像的均值, 并用这个空间均值代替整体均值。首先, 我们选择向量 \mathbf{m} 的所有元素等于图像 \mathbf{g} 的空间均值:

$$m_{kl} \equiv \mu_{\mathbf{g}} \equiv \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N g_{ij} \quad \forall k, l$$

变换后的图像 $\tilde{\mathbf{g}}$ 的元素给定如下:

$$\tilde{g}_{kl} = \sum_k A_{tk} (g_{kl} - m_{kl})$$

变换后的图像的均值是

$$\begin{aligned} \mu_{\tilde{\mathbf{g}}} &= \frac{1}{N^2} \sum_{t=1}^N \sum_{l=1}^N \tilde{g}_{tl} = \frac{1}{N^2} \sum_t \sum_l \sum_k A_{tk} (g_{kl} - \mu_{\mathbf{g}}) \\ &= \sum_k \left(\frac{1}{N} \sum_t A_{tk} \right) \left(\frac{1}{N} \sum_l (g_{kl} - \mu_{\mathbf{g}}) \right) \end{aligned}$$

在第一个括号内的数值是矩阵 A 的第 k 列的均值, 称为 a_k 。在第二个括号内的数值是原始图像减去均值后第 l 行的均值。可以看到, 变换后的图像具有0均值, 由矩阵 A 的列平均创建的向量和图像 \mathbf{g} 减去均值后的行的点积应为0。 A 是由图像 \mathbf{g} 的自协方差矩阵的特征向量所定义的, 所以它可能有这个性质。

我们所计算的自相关矩阵的另一个难题是, 它缺乏一些我们期望的性质: 它不是块循环的 (见6.31节“什么时候矩阵是块循环的?”), 并且它也不服从Wiener-Khinchine定理 (见第6章中的B6.2)。这样说的理由是由于我们计算时用到的定义: 假定图像存在是孤立的, 由大量的0围绕的。直观上这是正确的, 但它导致了自相关矩阵没有方便的结构。

109

为了使自相关矩阵是块循环的并且服从Wiener-Khinchine定理, 我们必须想像图像是自身重复的并且在各个方向都是无限的。所以, 我们可以利用下列公式计算它的自相关

矩阵:

$$C_{ij} = \frac{1}{N^2} \sum_{k=1}^N \sum_{l=1}^N g_{kl} g_{k-k_0, l-l_0}$$

其中

$$k_0 \equiv k_i - k_j = (i-1) \bmod N - (j-1) \bmod N$$

$$l_0 \equiv l_i - l_j = \frac{i-j-k_0}{N}$$

并且

若 $k-k_0 < 0$, 则用 $k-k_0+N$ 代替它

若 $l-l_0 < 0$, 则用 $l-l_0+N$ 代替它

若 $k-k_0 > N$, 则用 $k-k_0-N$ 代替它

若 $l-l_0 > N$, 则用 $l-l_0-N$ 代替它

注意, 这样的一个自相关矩阵会有很好的数学性质, 但它所揭示的关系可能不是真正存在的。例如: 点 (N, N) 和 $(1, 1)$ 的关系跟点 $(2, 2)$ 和 $(1, 1)$ 的关系是一样的, 点 $(N-1, N-1)$ 和 $(1, 1)$ 的关系跟点 $(3, 3)$ 和 $(1, 1)$ 的关系是一样的。

3.27 如何用K-L变换来逼近一幅图像

一幅图像的K-L变换给定如下:

$$\tilde{\mathbf{g}} = A(\mathbf{g} - \mu_{\mathbf{g}})$$

在这里 $\mu_{\mathbf{g}}$ 是一个 $N^2 \times 1$ 的向量, 这个向量的每一个元素都等于图像的平均灰度值, A 矩阵是由图像 \mathbf{g} 的自相关矩阵的特征向量所构成的, 将图像 \mathbf{g} 的自相关矩阵的特征值按降序排列, A 中的行是特征值对应的特征向量。反变换是:

$$\mathbf{g} = A^T \tilde{\mathbf{g}} + \mu_{\mathbf{g}}$$

如果我们将 \mathbf{g} 的自相关矩阵的最后几个特征值设为 0, 则矩阵 A 中对应的行将被替换为 0, 变换后的图像 $\tilde{\mathbf{g}}$ 中对应的行也为 0。

3.28 当我们截断K-L扩展来逼近一幅图像时的误差是怎样的

可以证明 (见方框 B3.3), 这种情况的逼近误差等于图像自协方差矩阵中所省略掉的特征值的和。

110

3.29 K-L变换的基图像是怎样的

因为 $\tilde{\mathbf{g}} = A(\mathbf{g} - \mu_{\mathbf{g}})$ 并且 A 是一个正交矩阵, 可得反变换为 $\mathbf{g} - \mu_{\mathbf{g}} = A^T \tilde{\mathbf{g}}$ 。可以很明确地写出这个表达式:

$$\begin{pmatrix} g_{11} - \mu_g \\ g_{21} - \mu_g \\ \vdots \\ g_{N1} - \mu_g \\ g_{21} - \mu_g \\ \vdots \\ g_{2N} - \mu_g \\ \vdots \\ g_{N1} - \mu_g \\ \vdots \\ g_{NN} - \mu_g \end{pmatrix} = \begin{pmatrix} a_{11} & a_{21} & \dots & a_{N1} \\ a_{12} & a_{22} & \dots & a_{N2} \\ \vdots & \vdots & & \vdots \\ a_{1N} & a_{2N} & \dots & a_{NN} \\ a_{1,N+1} & a_{2,N+1} & \dots & a_{N,N+1} \\ \vdots & \vdots & & \vdots \\ a_{1,2N} & a_{2,2N} & \dots & a_{N,2N} \\ \vdots & \vdots & & \vdots \\ a_{1,N^2-N+1} & a_{2,N^2-N+1} & \dots & a_{N,N^2-N+1} \\ \vdots & \vdots & & \vdots \\ a_{1,N^2} & a_{2,N^2} & \dots & a_{N,N^2} \end{pmatrix} \begin{pmatrix} \tilde{g}_{11} \\ \tilde{g}_{21} \\ \vdots \\ \tilde{g}_{N1} \\ \tilde{g}_{21} \\ \vdots \\ \tilde{g}_{2N} \\ \vdots \\ \tilde{g}_{N1} \\ \vdots \\ \tilde{g}_{NN} \end{pmatrix}$$

$$\Rightarrow \begin{cases} g_{11} - \mu_g = a_{11}\tilde{g}_{11} + a_{21}\tilde{g}_{21} + \dots + a_{N1}\tilde{g}_{NN} \\ g_{21} - \mu_g = a_{12}\tilde{g}_{11} + a_{22}\tilde{g}_{21} + \dots + a_{N2}\tilde{g}_{NN} \\ \dots \\ g_{N1} - \mu_g = a_{1N}\tilde{g}_{N1} + a_{2N}\tilde{g}_{21} + \dots + a_{NN}\tilde{g}_{NN} \\ g_{21} - \mu_g = a_{1,N+1}\tilde{g}_{N1} + a_{2,N+1}\tilde{g}_{21} + \dots + a_{N,N+1}\tilde{g}_{NN} \\ \dots \\ g_{2N} - \mu_g = a_{1,2N}\tilde{g}_{N1} + a_{2,2N}\tilde{g}_{21} + \dots + a_{N,2N}\tilde{g}_{NN} \\ \dots \\ g_{N1} - \mu_g = a_{1,N^2-N+1}\tilde{g}_{N1} + a_{2,N^2-N+1}\tilde{g}_{21} + \dots + a_{N,N^2-N+1}\tilde{g}_{NN} \\ \dots \\ g_{NN} - \mu_g = a_{1,N^2}\tilde{g}_{11} + a_{2,N^2}\tilde{g}_{21} + \dots + a_{N,N^2}\tilde{g}_{NN} \end{cases}$$

我们将这些方程重新整理成矩阵的形式:

$$\begin{pmatrix} g_{11} - \mu_g & g_{21} - \mu_g & \dots & g_{N1} - \mu_g \\ g_{21} - \mu_g & g_{22} - \mu_g & \dots & g_{N2} - \mu_g \\ \vdots & \vdots & & \vdots \\ g_{N1} - \mu_g & g_{2N} - \mu_g & \dots & g_{NN} - \mu_g \end{pmatrix} = \tilde{g}_{11} \begin{pmatrix} a_{11} & a_{1,N+1} & \dots & a_{1,N^2-N+1} \\ a_{12} & a_{1,N+2} & \dots & a_{1,N^2-N+2} \\ \vdots & \vdots & & \vdots \\ a_{1N} & a_{1,2N} & \dots & a_{1,N^2} \end{pmatrix} + \tilde{g}_{21} \begin{pmatrix} a_{21} & a_{2,N+1} & \dots & a_{2,N^2-N+1} \\ a_{22} & a_{2,N+2} & \dots & a_{2,N^2-N+2} \\ \vdots & \vdots & & \vdots \\ a_{2N} & a_{2,2N} & \dots & a_{2,N^2} \end{pmatrix} \\
 + \dots + \tilde{g}_{NN} \begin{pmatrix} a_{N1} & a_{N,N+1} & \dots & a_{N,N^2-N+1} \\ a_{N2} & a_{N,N+2} & \dots & a_{N,N^2-N+2} \\ \vdots & \vdots & & \vdots \\ a_{NN} & a_{N,2N} & \dots & a_{N,N^2} \end{pmatrix} \quad (3-47)$$

从这个表达式,可以看出根据K-L变换展开图像所用的特征图像是由图像的空间自相关矩阵的特征向量所形成的,只是将它们写成矩阵的形式。也就是说,通过将特征向量的前 N 个元素当做相应特征图像的第一列,接下来的 N 个元素当做下一列,对接下来的列也采用同样方法。这个表达式的系数是变换后图像的元素。

例3.11 考虑一个具有列表达式 g 的 3×3 图像。根据 g 的元素和变换矩阵 A 的元素 a_{ij} 写出图像的K-L变换的表达式。通过设置 A 的最后6行为0, 计算出图像 g 的一个近似解。证明这个近似解是一个 9×1 的向量, 其前3个元素等于 g 完全变换后的前3个元素, 而剩下的6个元素为零。

假设 μ_g 是图像 g 的平均灰度值。那么, 变换后的图像将有如下的形式:

$$\begin{aligned}
 \begin{pmatrix} \tilde{g}_{11} \\ \tilde{g}_{21} \\ \tilde{g}_{31} \\ \tilde{g}_{12} \\ \tilde{g}_{22} \\ \tilde{g}_{32} \\ \tilde{g}_{13} \\ \tilde{g}_{23} \\ \tilde{g}_{33} \end{pmatrix} &= \begin{pmatrix} a_{11} & a_{12} & \dots & a_{19} \\ a_{21} & a_{22} & \dots & a_{29} \\ a_{31} & a_{32} & \dots & a_{39} \\ a_{41} & a_{42} & \dots & a_{49} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ a_{91} & a_{92} & \dots & a_{99} \end{pmatrix} \begin{pmatrix} g_{11} - \mu_g \\ g_{21} - \mu_g \\ g_{31} - \mu_g \\ g_{12} - \mu_g \\ g_{22} - \mu_g \\ g_{32} - \mu_g \\ g_{13} - \mu_g \\ g_{23} - \mu_g \\ g_{33} - \mu_g \end{pmatrix} \\
 &= \begin{pmatrix} a_{11}(g_{11} - \mu_g) + a_{12}(g_{21} - \mu_g) + \dots + a_{19}(g_{33} - \mu_g) \\ a_{21}(g_{11} - \mu_g) + a_{22}(g_{21} - \mu_g) + \dots + a_{29}(g_{33} - \mu_g) \\ a_{31}(g_{11} - \mu_g) + a_{32}(g_{21} - \mu_g) + \dots + a_{39}(g_{33} - \mu_g) \\ a_{41}(g_{11} - \mu_g) + a_{42}(g_{21} - \mu_g) + \dots + a_{49}(g_{33} - \mu_g) \\ \vdots \\ a_{91}(g_{11} - \mu_g) + a_{92}(g_{21} - \mu_g) + \dots + a_{99}(g_{33} - \mu_g) \end{pmatrix} \quad (3-48) \quad \boxed{112}
 \end{aligned}$$

如果设 $a_{41} = a_{42} = \dots = a_{49} = a_{51} = \dots = a_{59} = \dots = a_{99} = 0$, 很明显, 上述向量的最后6行为0, 而截断后的变换图像将是向量:

$$\tilde{g}' = (\tilde{g}_{11} \quad \tilde{g}_{21} \quad \tilde{g}_{31} \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0)^T \quad (3-49)$$

根据式(3-47)图像的近似解为

$$\begin{aligned}
 \begin{pmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{pmatrix} &= \begin{pmatrix} \mu_g & \mu_g & \mu_g \\ \mu_g & \mu_g & \mu_g \\ \mu_g & \mu_g & \mu_g \end{pmatrix} + \tilde{g}_{11} \begin{pmatrix} a_{11} & a_{14} & a_{17} \\ a_{12} & a_{15} & a_{18} \\ a_{13} & a_{16} & a_{19} \end{pmatrix} \\
 &+ \tilde{g}_{21} \begin{pmatrix} a_{21} & a_{24} & a_{27} \\ a_{22} & a_{25} & a_{28} \\ a_{23} & a_{26} & a_{29} \end{pmatrix} + \tilde{g}_{31} \begin{pmatrix} a_{31} & a_{34} & a_{37} \\ a_{32} & a_{35} & a_{38} \\ a_{33} & a_{36} & a_{39} \end{pmatrix} \quad (3-50)
 \end{aligned}$$

例3.12(B) 证明: 如果 A 是一个 $N^2 \times N^2$ 的矩阵, 它的第 i 行是向量 u_i^T 。 C_2 是一个 $N^2 \times N^2$ 的矩阵, 它除了位置(2,2)的元素等于 C_2 以外, 其他元素都为0。那么

$$A^T C_2 A = c_2 u_2 u_2^T$$

假设 u_{ij} 表示向量 u_i 的第 j 个元素。那么

$$\begin{aligned}
A^T C_2 A &= \begin{pmatrix} u_{11} & u_{21} & \dots & u_{N^2 1} \\ u_{12} & u_{22} & \dots & u_{N^2 2} \\ u_{13} & u_{23} & \dots & u_{N^2 3} \\ \vdots & \vdots & & \vdots \\ u_{1N^2} & u_{2N^2} & \dots & u_{N^2 N^2} \end{pmatrix} \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & c_2 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} \\
&= \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1N^2} \\ u_{21} & u_{22} & \dots & u_{2N^2} \\ u_{31} & u_{32} & \dots & u_{3N^2} \\ \vdots & \vdots & & \vdots \\ u_{N^2 1} & u_{N^2 2} & \dots & u_{N^2 N^2} \end{pmatrix} \begin{pmatrix} 0 & 0 & \dots & 0 \\ c_2 u_{21} & c_2 u_{22} & \dots & c_2 u_{2N^2} \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} \\
&= \begin{pmatrix} c_2 u_{21}^2 & c_2 u_{21} u_{22} & \dots & c_2 u_{21} u_{2N^2} \\ c_2 u_{22} u_{21} & c_2 u_{22}^2 & \dots & c_2 u_{22} u_{2N^2} \\ \vdots & \vdots & & \vdots \\ c_2 u_{2N^2} u_{21} & c_2 u_{2N^2} u_{22} & \dots & c_2 u_{2N^2}^2 \end{pmatrix} \\
&= c_2 \mathbf{u}_2 \mathbf{u}_2^T
\end{aligned} \tag{3-51}$$

例3.13(B) 假设一个 3×3 的图像，我们用它自协方差矩阵的前三个特征值去逼近它。证明：

$$E\{\tilde{\mathbf{g}}\tilde{\mathbf{g}}'^T\} = C' \tilde{\mathbf{g}}\tilde{\mathbf{g}} \tag{3-52}$$

利用例3.12中关于截断图像 $\tilde{\mathbf{g}}'$ 的变换的结果，我们有：

$$\begin{aligned}
E\{\tilde{\mathbf{g}}\tilde{\mathbf{g}}'^T\} &= E \left\{ \begin{pmatrix} \tilde{g}_{11} \\ \tilde{g}_{21} \\ \tilde{g}_{31} \\ \tilde{g}_{12} \\ \tilde{g}_{22} \\ \tilde{g}_{32} \\ \tilde{g}_{13} \\ \tilde{g}_{23} \\ \tilde{g}_{33} \end{pmatrix} (\tilde{g}_{11} \ \tilde{g}_{21} \ \tilde{g}_{31} \ 0 \ 0 \ 0 \ 0 \ 0 \ 0) \right\} \\
&= E \left\{ \begin{pmatrix} \tilde{g}_{11}^2 & \tilde{g}_{11}\tilde{g}_{21} & \tilde{g}_{11}\tilde{g}_{31} & 0 & 0 & 0 & 0 & 0 & 0 \\ \tilde{g}_{21}\tilde{g}_{11} & \tilde{g}_{21}^2 & \tilde{g}_{21}\tilde{g}_{31} & 0 & 0 & 0 & 0 & 0 & 0 \\ \tilde{g}_{31}\tilde{g}_{11} & \tilde{g}_{31}\tilde{g}_{21} & \tilde{g}_{31}^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{g}_{33}\tilde{g}_{11} & \tilde{g}_{33}\tilde{g}_{21} & \tilde{g}_{33}\tilde{g}_{31} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \right\}
\end{aligned}$$

$$= \begin{pmatrix} E\{\tilde{g}_{11}^2\} & E\{\tilde{g}_{11}\tilde{g}_{21}\} & E\{\tilde{g}_{11}\tilde{g}_{31}\} & 0 & 0 & 0 & 0 & 0 & 0 \\ E\{\tilde{g}_{21}\tilde{g}_{11}\} & E\{\tilde{g}_{21}^2\} & E\{\tilde{g}_{21}\tilde{g}_{31}\} & 0 & 0 & 0 & 0 & 0 & 0 \\ E\{\tilde{g}_{31}\tilde{g}_{11}\} & E\{\tilde{g}_{31}\tilde{g}_{21}\} & E\{\tilde{g}_{31}^2\} & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & & & & & & & & \\ E\{\tilde{g}_{33}\tilde{g}_{11}\} & E\{\tilde{g}_{33}\tilde{g}_{21}\} & E\{\tilde{g}_{33}\tilde{g}_{31}\} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

变换后的图像 $\tilde{\mathbf{g}}$ 由一个矩阵构成, 它的相关矩阵的非对角线元素都为0。所以, 有

$$E\{\tilde{\mathbf{g}}\tilde{\mathbf{g}}'^T\} = \begin{pmatrix} E\{\tilde{g}_{11}^2\} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & E\{\tilde{g}_{21}^2\} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & E\{\tilde{g}_{31}^2\} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} = C'_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}} \quad \vdots$$

B3.3 用K-L变换来逼近图像的误差是多大?

我们将证明, K-L变换不只以不相关的数据来表示图像, 而且, 如果在某项后截断变换, 它可用于在最小均方误差的意义下逼近图像。设变换为:

$$\tilde{\mathbf{g}} = \mathbf{A}\mathbf{g} - \mathbf{A}\boldsymbol{\mu}_{\mathbf{g}} \Rightarrow \mathbf{g} = \mathbf{A}^T\tilde{\mathbf{g}} + \boldsymbol{\mu}_{\mathbf{g}} \quad (3-53)$$

假设将 $C_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}}$ 的特征值按降序排列。忽略最后的几个特征值, 只保留前面 K 个比较重要的特征值。 $C_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}}$ 也是 $N^2 \times N^2$ 的矩阵, 它截断后为 $C'_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}}$, 最后 $N^2 - K$ 个对角元素为0。变换矩阵 \mathbf{A}^T 也是 $N^2 \times N^2$ 的矩阵, 它的列是由 $C_{\tilde{\mathbf{g}}\tilde{\mathbf{g}}}$ 的特征向量所组成。忽略 $N^2 - K$ 个特征值也就是忽略 $N^2 - K$ 个特征向量, 所以新的变换矩阵 \mathbf{A}^T 最后的 $N^2 - K$ 个列为0。近似的图像为:

$$\mathbf{g}' = \mathbf{A}^T\tilde{\mathbf{g}}' + \boldsymbol{\mu}_{\mathbf{g}} \quad (3-54)$$

图像的近似误差为 $\mathbf{g} - \mathbf{g}' = \mathbf{A}^T\tilde{\mathbf{g}} - \mathbf{A}^T\tilde{\mathbf{g}}'$ 。矩阵的范数为:

$$\|\mathbf{g} - \mathbf{g}'\| = \text{trace}[(\mathbf{g} - \mathbf{g}')(\mathbf{g} - \mathbf{g}')^T] \quad (3-55)$$

所以, 均方误差为:

$$E\{\|\mathbf{g} - \mathbf{g}'\|\} = E\{\text{trace}[(\mathbf{g} - \mathbf{g}')(\mathbf{g} - \mathbf{g}')^T]\} \quad (3-56)$$

由于取迹只是将矩阵的对角元素相加起来, 所以可以交换取迹(trace)跟期望的顺序:

$$\begin{aligned} E\{\|\mathbf{g} - \mathbf{g}'\|\} &= \text{trace}\left[E\{(\mathbf{g} - \mathbf{g}')(\mathbf{g} - \mathbf{g}')^T\}\right] \\ &= \text{trace}\left[E\left\{(\mathbf{A}^T\tilde{\mathbf{g}} - \mathbf{A}^T\tilde{\mathbf{g}}')(\mathbf{A}^T\tilde{\mathbf{g}} - \mathbf{A}^T\tilde{\mathbf{g}}')^T\right\}\right] \\ &= \text{trace}\left[E\left\{(\mathbf{A}^T\tilde{\mathbf{g}} - \mathbf{A}^T\tilde{\mathbf{g}}')(\tilde{\mathbf{g}}^T\mathbf{A} - \tilde{\mathbf{g}}'^T\mathbf{A}')\right\}\right] \\ &= \text{trace}\left[E\left\{\mathbf{A}^T\tilde{\mathbf{g}}\tilde{\mathbf{g}}^T\mathbf{A} - \mathbf{A}^T\tilde{\mathbf{g}}\tilde{\mathbf{g}}'^T\mathbf{A}' - \mathbf{A}'^T\tilde{\mathbf{g}}'\tilde{\mathbf{g}}^T\mathbf{A} + \mathbf{A}'^T\tilde{\mathbf{g}}'\tilde{\mathbf{g}}'^T\mathbf{A}'\right\}\right] \end{aligned} \quad (3-57)$$

矩阵 A 和 A' 是固定的, 因此期望算子对它们没有影响。所以

$$E\{\|g - g'\|\} = \text{trace}[A^T E\{\tilde{g}\tilde{g}^T\}A - A^T E\{\tilde{g}\tilde{g}'^T\}A' - A'^T E\{\tilde{g}'\tilde{g}^T\}A + A'^T E\{\tilde{g}'\tilde{g}'^T\}A'] \quad (3-58)$$

在这个表达式中, 我们可以将 $E\{\tilde{g}\tilde{g}^T\}$ 和 $E\{\tilde{g}'\tilde{g}'^T\}$ 看做是两个变换: $C_{\tilde{g}\tilde{g}}$ 和 $C'_{\tilde{g}\tilde{g}}$ 的相关矩阵。

$\tilde{g}\tilde{g}'^T$ 矩阵是一个向量和它自己转置的积, 而且这个转置的最后 N^2-K 个元素都为0。期望算子将 $\tilde{g}\tilde{g}'$ 的所有非对角线元素变为0 (因为变换的自相关矩阵除了对角线元素外都为0)。事实上 \tilde{g}'^T 的最后 N^2-K 个元素也为0 (见例3.13)。所以, 结果是

$$E\{\tilde{g}\tilde{g}'^T\} = C'_{\tilde{g}\tilde{g}} \quad (3-59)$$

同理可得

$$E\{\tilde{g}'\tilde{g}^T\} = C'_{\tilde{g}\tilde{g}} \quad (3-60)$$

所以

$$E\{\|g - g'\|\} = \text{trace}[A^T C_{\tilde{g}\tilde{g}} A - A^T C'_{\tilde{g}\tilde{g}} A' - A'^T C'_{\tilde{g}\tilde{g}} A + A'^T C'_{\tilde{g}\tilde{g}} A'] \quad (3-61)$$

考虑和: $-A^T C'_{\tilde{g}\tilde{g}} A' + A'^T C'_{\tilde{g}\tilde{g}} A = -(A - A')^T C'_{\tilde{g}\tilde{g}} A'$, 我们可以将 A 分割成两个部分, $K \times N^2$ 的子矩阵 A_1 和 $(N^2-K) \times N^2$ 的子矩阵 A_2 。 A' 由 A_1 和所有元素为0的 $(N^2-K) \times N^2$ 子矩阵组成。

$$A = \begin{pmatrix} A_1 \\ - \\ A_2 \end{pmatrix} \quad A' = \begin{pmatrix} A_1 \\ - \\ 0 \end{pmatrix} \Rightarrow$$

$$A - A' = \begin{pmatrix} 0 \\ - \\ A_2 \end{pmatrix} \quad \text{和} \quad (A - A')^T = \left(\underbrace{0}_{N^2 \times K} \mid \underbrace{A_2^T}_{N^2 \times (N^2-K)} \right) \quad (3-62)$$

那么 $(A - A')^T C'_{\tilde{g}\tilde{g}} A' = (0 \mid A_2^T) C'_{\tilde{g}\tilde{g}} A'$ 。

$C'_{\tilde{g}\tilde{g}}$ 能分成四个子矩阵:

$$C'_{\tilde{g}\tilde{g}} = \begin{pmatrix} C_1 & \mid & 0 \\ - & - & - \\ 0 & \mid & 0 \end{pmatrix} \quad (3-63)$$

这里 C_1 是一个 $K \times K$ 的对角阵。那么积是:

$$(0 \mid A_2^T) \begin{pmatrix} C_1 & \mid & 0 \\ - & - & - \\ 0 & \mid & 0 \end{pmatrix} = (0) \quad (3-64)$$

利用式(3-61)我们得到的结果:

$$E\{\|g - g'\|\} = \text{trace}[A^T C_{\tilde{g}\tilde{g}} A - A'^T C'_{\tilde{g}\tilde{g}} A] \quad (3-65)$$

考虑项 $A^T C_{gg} A$ 。我们可以假定 C_{gg} 是 N^2 个矩阵的和, 每一个矩阵都是 $N^2 \times N^2$ 的, 而且只有一个非0元素:

$$C_{gg} = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} + \dots + \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \lambda_{N^2} \end{pmatrix} \quad (3-66)$$

A 是由特征向量行组成的, 而且 A^T 是由特征向量列组成的。所以我们可以写成:

$$A^T C_{gg} A = \sum_{i=1}^{N^2} (\mathbf{u}_1 \quad \mathbf{u}_2 \quad \dots \quad \mathbf{u}_{N^2}) C_i \begin{pmatrix} \mathbf{u}_1^T \\ \mathbf{u}_2^T \\ \vdots \\ \mathbf{u}_{N^2}^T \end{pmatrix} \quad (3-67)$$

这里, 矩阵 C 的第 i 个对角元素是非0且等于 λ_i 。由例3.11的结果, 我们有:

$$\begin{aligned} \text{trace}[A^T C_{gg} A] &= \text{trace} \left[\sum_{i=1}^{N^2} \lambda_i \mathbf{u}_i \mathbf{u}_i^T \right] \\ &= \text{trace} \left[\sum_{i=1}^{N^2} \lambda_i \begin{pmatrix} u_{i1}^2 & u_{i1}u_{i2} & \dots & u_{i1}u_{iN^2} \\ u_{i2}u_{i1} & u_{i2}^2 & \dots & u_{i2}u_{iN^2} \\ \vdots & \vdots & & \vdots \\ u_{iN^2}u_{i1} & u_{iN^2}u_{i2} & \dots & u_{iN^2}^2 \end{pmatrix} \right] \\ &= \sum_{i=1}^{N^2} \lambda_i \text{trace} \begin{pmatrix} u_{i1}^2 & u_{i1}u_{i2} & \dots & u_{i1}u_{iN^2} \\ u_{i2}u_{i1} & u_{i2}^2 & \dots & u_{i2}u_{iN^2} \\ \vdots & \vdots & & \vdots \\ u_{iN^2}u_{i1} & u_{iN^2}u_{i2} & \dots & u_{iN^2}^2 \end{pmatrix} \\ &= \sum_{i=1}^{N^2} \lambda_i (u_{i1}^2 + u_{i2}^2 + \dots + u_{iN^2}^2) \\ &= \sum_{i=1}^{N^2} \lambda_i \end{aligned} \quad (3-68)$$

为了得到上面的结果, 我们利用了事实: 由于 \mathbf{u}_i 是特征向量, 所以有 $u_{i1}^2 + u_{i2}^2 + \dots + u_{iN^2}^2 = 1$, 将这个结果应用到式(3-65), 得到:

$$\text{均方误差} = \sum_{i=1}^{N^2} \lambda_i - \sum_{i=1}^K \lambda_i = \sum_{i=K+1}^{N^2} \lambda_i \quad (3-69)$$

所以, 当一个图像由它的截断K-L变换来近似的时候, 均方误差等于相关矩阵所忽略的特征值的和。由于 λ_i 是降序排列的, 这也证明了均方误差是最小的。

117

118

• 例3.14 一个 2×2 图像的自相关矩阵给定如下:

$$C = \begin{pmatrix} 3 & 0 & -1 & 0 \\ 0 & 3 & 0 & -1 \\ -1 & 0 & 3 & 0 \\ 0 & -1 & 0 & 3 \end{pmatrix}$$

计算出图像的变换矩阵 A , 该变换矩阵将用在反变换中近似均方误差为2的图像。

解下列方程, 可得到矩阵的特征值:

$$\begin{aligned} & \begin{vmatrix} 3-\lambda & 0 & -1 & 0 \\ 0 & 3-\lambda & 0 & -1 \\ -1 & 0 & 3-\lambda & 0 \\ 0 & -1 & 0 & 3-\lambda \end{vmatrix} = 0 \Rightarrow \\ & (3-\lambda) [(3-\lambda)^3 - (3-\lambda)] - (-1)^2 [(3-\lambda)^2 - (-1)^2] = 0 \Rightarrow \\ & (3-\lambda)^2 [(3-\lambda)^2 - 1] - [(3-\lambda)^2 - 1] = 0 \Rightarrow \\ & [(3-\lambda)^2 - 1]^2 = 0 \Rightarrow \\ & (3-\lambda-1)^2 (3-\lambda+1)^2 = 0 \Rightarrow \\ & (2-\lambda)^2 (4-\lambda)^2 = 0 \Rightarrow \\ & \lambda_1 = 4, \lambda_2 = 4, \lambda_3 = 2, \lambda_4 = 2 \end{aligned} \quad (3-70)$$

特征值 $\lambda = 4$ 的特征向量为:

$$\begin{aligned} & \begin{pmatrix} 3 & 0 & -1 & 0 \\ 0 & 3 & 0 & -1 \\ -1 & 0 & 3 & 0 \\ 0 & -1 & 0 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = 4 \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \Rightarrow \begin{cases} 3x_1 - x_3 = 4x_1 \\ 3x_2 - x_4 = 4x_2 \\ -x_1 + 3x_3 = 4x_3 \\ -x_2 + 3x_4 = 4x_4 \end{cases} \Rightarrow \\ & x_3 = -x_1, x_4 = -x_2, x_1 = -x_3, x_2 = -x_4 \end{aligned} \quad (3-71)$$

选择: $x_1 = x_3 = 0, x_2 = \frac{1}{\sqrt{2}}, x_4 = -\frac{1}{\sqrt{2}}$

或选择: $x_1 = \frac{1}{\sqrt{2}}, x_3 = -\frac{1}{\sqrt{2}}, x_2 = x_4 = 0$

所以, 前两个特征向量为: $(0 \ \frac{1}{\sqrt{2}} \ 0 \ -\frac{1}{\sqrt{2}})$ 和 $(\frac{1}{\sqrt{2}} \ 0 \ -\frac{1}{\sqrt{2}} \ 0)$

119

它们是正交的, 对于特征值 $\lambda = 2$ 我们有:

$$\begin{aligned} & \begin{pmatrix} 3 & 0 & -1 & 0 \\ 0 & 3 & 0 & -1 \\ -1 & 0 & 3 & 0 \\ 0 & -1 & 0 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = 2 \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \Rightarrow \begin{cases} 3x_1 - x_3 = 2x_1 \\ 3x_2 - x_4 = 2x_2 \\ -x_1 + 3x_3 = 2x_3 \\ -x_2 + 3x_4 = 2x_4 \end{cases} \Rightarrow \\ & x_1 = x_3, x_2 = x_4, x_1 = x_3, x_2 = x_4 \end{aligned} \quad (3-72)$$

选择: $x_1 = x_3 = 0, x_2 = x_4 = \frac{1}{\sqrt{2}}$

我们不需要计算第4个特征向量，因为我们只对近似变换矩阵感兴趣。通过设置一些特征向量为(0 0 0 0)，重构图像的均方误差等于相应的特征值的和。在这种情况下，如果我们的变换矩阵为矩阵 \tilde{A} ：

$$\tilde{A} = \begin{pmatrix} 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (3-73)$$

误差为 $\lambda_4 = 2$

例3.15 显示例2.14中的图像的K-L变换的不同层次。

图3-2显示了例2.14中图像根据K-L变换得到的64幅特征图像。

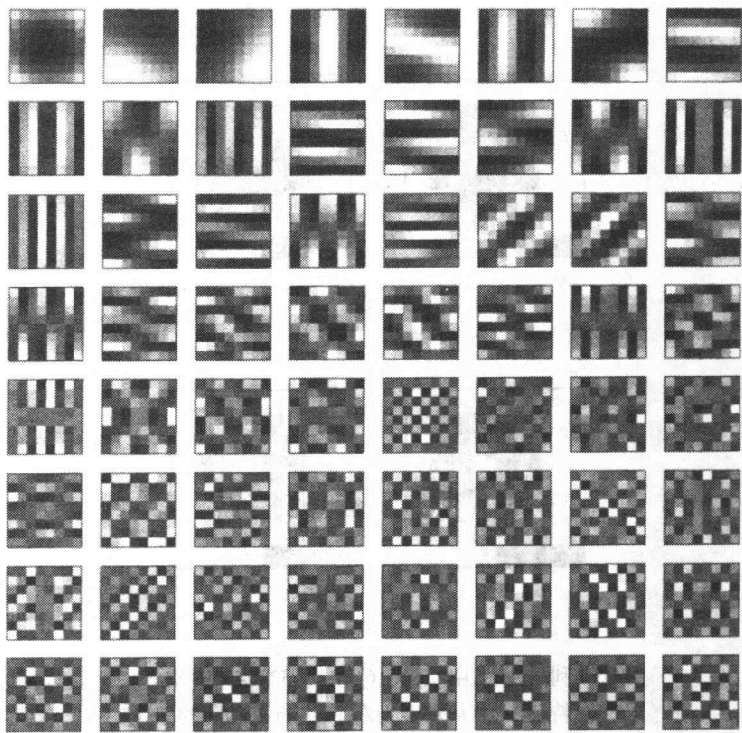


图3-2 64个特征图像，每幅图像的值在0~255之间

在图3-3中的8幅图像是重构后的图像，是分别用8、16、24、32、40、48、56和64幅特征图像进行重构的。

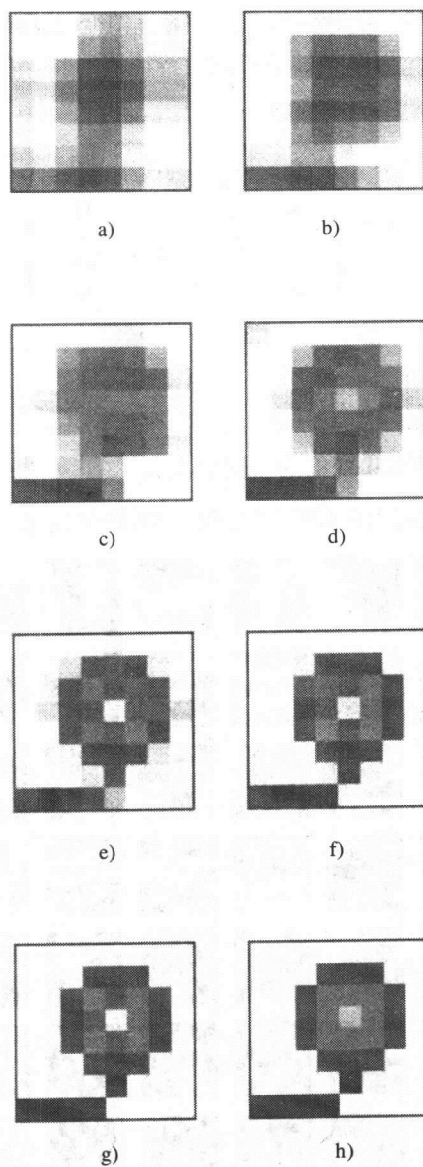


图3-3 利用图3-2中前8、16、24、32、40、48、56和64幅特征图像(从左上到右下)进行重构后的图像

每一幅重构图像的均方误差的和是:

图3-3a的误差: $5360(\sum_{i=9}^{64} \lambda_i = 5365)$

图3-3b的误差: $3846(\sum_{i=17}^{64} \lambda_i = 3850)$

图3-3c的误差: $2715(\sum_{i=25}^{64} \lambda_i = 2718)$

图3-3d的误差: $1850(\sum_{i=33}^{64} \lambda_i = 1852)$

图3-3e的误差: $1194(\sum_{i=41}^{64} \lambda_i = 1195)$

图3-3f的误差: $715(\sum_{i=49}^{64} \lambda_i = 715)$

图3-3g的误差: $321(\sum_{i=57}^{64} \lambda_i = 321)$

图3-3h的误差: 0

注意, 重建后图像的均方误差跟所忽略的特征值的和很接近。

120
122

例3.16 给定 2×2 图像的自相关矩阵如下:

$$C = \begin{pmatrix} 4 & 0 & -1 & 0 \\ 0 & 4 & 0 & -1 \\ -1 & 0 & 4 & 0 \\ 0 & -1 & 0 & 4 \end{pmatrix}$$

计算图像的变换矩阵A, 当用在逆变换时, 近似图像的均方误差等于6。

首先, 找出矩阵的特征值:

$$\begin{aligned} & \begin{vmatrix} 4-\lambda & 0 & -1 & 0 \\ 0 & 4-\lambda & 0 & -1 \\ -1 & 0 & 4-\lambda & 0 \\ 0 & -1 & 0 & 4-\lambda \end{vmatrix} = 0 \Rightarrow \\ (4-\lambda) & \begin{vmatrix} 4-\lambda & 0 & -1 \\ 0 & 4-\lambda & 0 \\ -1 & 0 & 4-\lambda \end{vmatrix} - 1 \begin{vmatrix} 0 & 4-\lambda & -1 \\ -1 & 0 & 0 \\ 0 & -1 & 4-\lambda \end{vmatrix} = 0 \Rightarrow \\ & (4-\lambda)[(4-\lambda)^2 - (4-\lambda)] - [(4-\lambda)^2 - 1] = 0 \Rightarrow \\ & [(4-\lambda)^2 - 1]^2 = 1 \Rightarrow (4-\lambda-1)^2(4-\lambda+1)^2 = 0 \Rightarrow \\ & \lambda_1 = 5 \\ & \lambda_2 = 5 \\ & \lambda_3 = 3 \\ & \lambda_4 = 3 \end{aligned}$$

因为允许重构图像后的误差为6, 所以不需要计算特征值 $\lambda = 3$ 所对应的特征向量。

特征值 $\lambda = 5$ 的特征向量:

$$\begin{pmatrix} 4 & 0 & -1 & 0 \\ 0 & 4 & 0 & -1 \\ -1 & 0 & 4 & 0 \\ 0 & -1 & 0 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = 5 \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \Rightarrow \begin{aligned} 4x_1 - x_3 &= 5x_1 \Rightarrow x_1 = x_3 \\ 4x_2 - x_4 &= 5x_2 \Rightarrow x_2 = -x_4 \\ -x_1 - 4x_3 &= 5x_3 \Rightarrow x_1 = -x_3 \\ -x_2 - 4x_4 &= 5x_4 \Rightarrow x_2 = -x_4 \end{aligned}$$

选择 $x_1 = x_3 = 0$, $x_2 = \frac{1}{\sqrt{2}}$, $x_4 = -\frac{1}{\sqrt{2}}$ 。对于 λ_2 选择一下正交的特征向量, 也就是 $x_2 = x_4 = 0$, $x_1 = \frac{1}{\sqrt{2}}$, $x_3 = -\frac{1}{\sqrt{2}}$ 。那么允许重构后的误差(等于所忽略的特征值的和)为6的变换矩阵为:

123

$$A = \begin{pmatrix} 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

•
•
•

3.30 本章要点

如果将一幅图像看做随机过程中所产生的图像集中的一个实例, 我们也可以用特征图像的线性叠加来表示, 而这些特征图像适合表示图像集中任意一幅。对于一个 $N \times N$ 的图像, 它有多达 N^2 个这样的特征图像, 但如果用奇异值分解(SVD)的方法来做, 我们只需要 N 幅特征图像。这两种方法的不同点在于, 用 N^2 个特征图像的时候, 我们可以表示图像集里面的所有图像, 而用 N 幅特征图像的时候, 我们只能表示一幅图像。如果特征图像按照对应特征值的降序排列, 截断它的某些项, 就可以在整个图像集上面, 以最小均方误差的要求近似地表示原图像。在奇异值分解的情况下, 截短可以产生最小均方误差的近似。

K-L展开的难题在于遍历性的假设。这个假设要求, 一幅图像的空间统计性质跟图像集上面的空间统计性质是一样的。如果考虑某类限定的图像, 这个假设很明显是不成立的: 图像不是一个随机过程输出的简单结果; 总是有一个潜在的确定性部分使这个假设不成立。所以, 在这种情况下, K-L变换将更多的重点放在图像的随机部分上面。如噪声, 而不是感兴趣的部分。尽管如此, 如果考虑很多幅不同图像的话, 在图像集上面的每一个像素点的平均灰度值都是相同的, 遍历性的假设也是近似成立的。另外, 如果图像集表示某类感兴趣的图像, 遍历性的假设也就没有必要了: K-L变换可以通过整体的统计计算出来, 可用于定义某类特殊的图像的泰勒(taylor-made)展开基图像。

124

第4章 图像增强

4.1 什么是图像增强

图像增强是指改善图像,以使图像主观上看起来更好的一种图像处理方法。虽然我们并不关心一幅图像的内容是什么,但是我们关心这副图像是否可以得到改善。比如,是否能观察到更多的细节;是否能去掉一些不想要的斑点;是否能使对比度达到更好?等等。

4.2 如何增强图像

图像增强的方法主要取决于图像希望达到的特定效果。一般来说,图像增强的方法分为两大类:基于图像灰度值统计的方法和基于图像空间频率的方法。

4.3 利用图像的灰度级统计进行图像增强有哪些方法

- 用直方图处理图像以增强对比度。
- 用主分量分析处理多光谱图像以获得最大对比度。
- 用排序滤波器去除图像噪声。

4.4 什么是图像的直方图

图像的直方图是一个关于图像灰度级别的离散函数,描述的是图像中具有该灰度级别的像素的个数。直方图归一化之后,该函数在所有灰度级上的离散和为1,此时,该离散函数可看成是图像中具有相同灰度值的像素出现的概率密度函数。按照这个方式,一个像素的灰度值就是在随机试验过程中的一个随机变量。

125

4.5 什么时候有必要修改图像的直方图

假设一幅图像不能观察到很多细节。原因很可能是图像中不同的对象或同一对象的不同部分拥有相同或相近的灰度值。从图4-1的例子可以看到,“差”图像的直方图很窄,而“好”图像的直方图则要平坦许多。

4.6 如何修改图像的直方图

假设原图图像灰度值变量为 r ,变换后的新图的灰度值变量为 s 。我们希望找到一个变换 $s = T(r)$,使得图4-1a中的概率密度函数 $p_r(r)$ 变换成图4-1b中的概率密度函数 $p_s(s)$ 。

由于 $p_r(r)$ 是随机变量为 r 的概率密度函数,原图灰度值在区间 $[r, r + dr]$ 上的像素的个数为 $p_r(r)dr$ 。又因为变换使得原区间 $[r, r + dr]$ 变换为新区间 $[s, s + ds]$,而且像素总数在该区间保持不变,所以在增强后的图像中,灰度值在区间 $[s, s + ds]$ 上的像素个数为 $p_s(s)ds$ 。于是有

126

$$p_s(s)ds = p_r(r)dr \quad (4-1)$$

这个等式表明, 给定函数 $p_s(s)$, 就可定义变量为 r 的变换 $s = T(r)$ 。

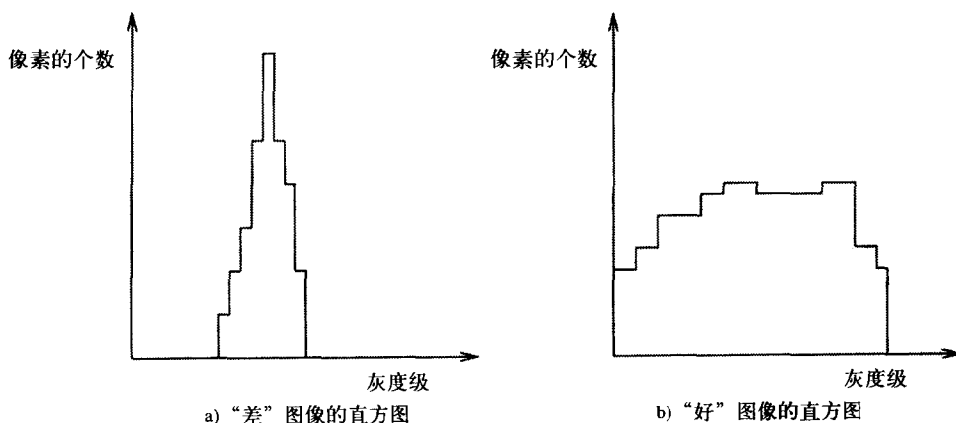


图 4-1

4.7 什么是直方图均衡化

直方图均衡化是对图像进行处理, 使得图像直方图中所有的灰度级别尽量拥有相同的像素个数。也就是说, 直方图均衡化使得 $p_s(s) = c$, 其中 c 为常数。把 $p_s(s)$ 代入式(4-1)并两边积分可得变换 $s = T(r)$ 。由于式(4-1)恒成立, 于是可以对式(4-1)两边从0到任意值积分, 这等价于两个随机变量 s 和 r 的分布函数相等。

$$\int_0^s cds = \int_0^r p_r(r)dr \Rightarrow s = \frac{1}{c} \int_0^r p_r(x)dx \quad (4-2)$$

这里, 为了避免混淆, 我们用 x 代替上述积分变量。图4-2a ~ 图4-2d显示了一个对小对比度图像进行直方图均衡化的例子。注意, 原图4-2a的直方图4-2b很窄。而执行直方图均衡化之后, 直方图4-2d拉伸了许多, 但直方图并没有预料中那样平坦。也就是说, 直方图看起来没那么“均衡”。

4.8 为什么直方图均衡化并不会产生具有平坦直方图的图像

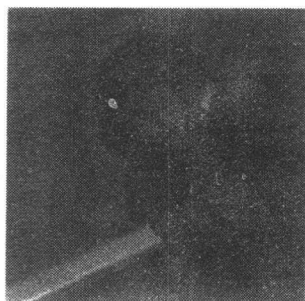
在上面的分析中, 我们假定变量 s 和 r 是连续的。然而在实际中, 灰度级却是离散的。在连续形式下, 任意区间间隔 $[r, r + dr]$ 上都有无穷个数。而在数字图像中, 每个灰度值区间范围内却仅有有限个像素。灰度区间拉伸时, 区间内的像素并没有增加。所以与连续形式不同, 一般离散变换不能产生均匀概率密度的离散值, 但直方图均衡化以后, 可看出直方图有趋于平坦的趋势, 像素的灰度值几乎能分布在整個灰度区间, 但直方图并不平坦。

4.9 增强图像使它具有绝对平坦的直方图是否可能

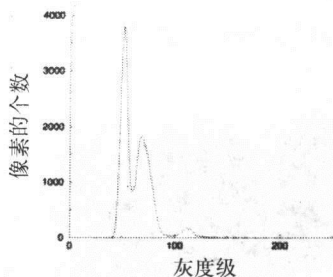
回答是肯定的, 这只需在相邻灰度级中重新随机分布像素。这种方法称为随机增量(random addition)式直方图修正法。考虑一个简单的例子, 设原图中有 N_1 个灰度值为 g_1 的像

素,有 N_2 个灰度值为 g_2 的像素,进行图像增强的目的是拉伸直方图,使得增强的图像中灰度值为 \tilde{g}_1 、 \tilde{g}_2 和 \tilde{g}_3 的像素分别有 $(N_1 + N_2)/3$ 个,而且变换时假定把 g_i 变换成 \tilde{g}_i 。事实上,在进行图像变换后,我们发现灰度值为 \tilde{g}_1 、 \tilde{g}_2 和 \tilde{g}_3 的像素个数分别为 \tilde{N}_1 、 \tilde{N}_2 和 \tilde{N}_3 ,其中 $\tilde{N}_1 > (N_1 + N_2)/3$ 、 $\tilde{N}_2 < (N_1 + N_2)/3$ 、 $\tilde{N}_3 < (N_1 + N_2)/3$ 。此时,我们可以随机取出 $(N_1 + N_2)/3 - \tilde{N}_3$ 个灰度值为 \tilde{g}_2 的像素,使其变换成灰度值为 \tilde{g}_3 的像素。接着,我们随机取出 $\tilde{N}_1 - (N_1 + N_2)/3$ 个灰度值为 \tilde{g}_1 的像素,使其变换成灰度值为 \tilde{g}_2 的像素。因此,图像增强能使其具有绝对平坦的直方图。图4-2e和图4-3c就是应用随机增量式直方图修正法的例子。

127



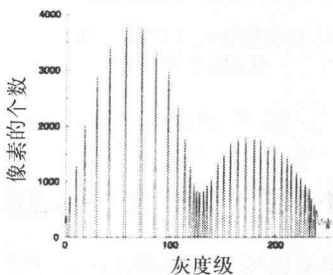
a) 原图



b) 原图的直方图



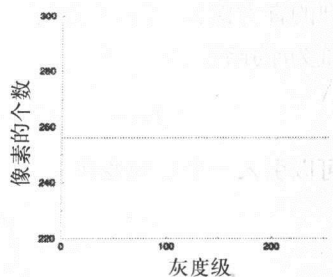
c) 直方图均衡化后的图像



d) 图像直方图均衡化后的直方图



e) 随机增量式直方图修正法处理后的图像



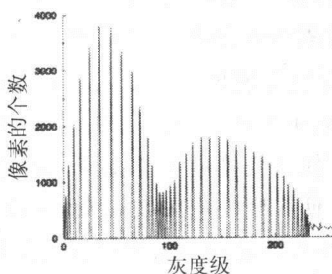
f) 随机增量式直方图修正法处理后的直方图

图4-2 用直方图均衡化进行图像增强

128



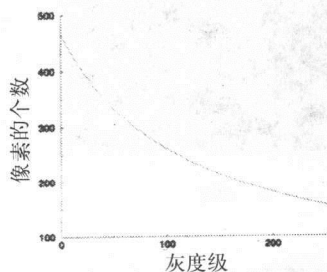
a) 直方图夸张化后的图像



b) 图像直方图夸张化后的直方图



c) 随机增量式直方图夸张化处理后的图像



d) 随机增量式直方图夸张化处理后的直方图

图4-3 对图4-2a的图像进行参数 $\alpha = 0.5$ 的直方图夸张化

4.10 如果我们不想要一幅有绝对平坦直方图的图像应该怎么做

我们可以任意定义概率密度函数 $p_s(s)$ ，于是

$$\int_0^s p_s(y) dy = \int_0^r p_r(x) dx$$

129

这里，为了避免混淆，我们用 x 和 y 代替上述积分变量。上式决定 s 随 r 的变化关系。由于 $p_s(y)$ 已知（希望得到的直方图），等式左边可改写为一个变量为 s 的函数 $f_1(s)$ ，类似地，等式右边可改写为一个变量为 r 的函数 $f_2(r)$ 。于是

$$f_1(s) = f_2(r) \Rightarrow s = f_1^{-1} f_2(r) \quad (4-3)$$

实际上，我们可以引入一个中间变换

$$w \equiv \int_0^r p_r(x) dx \quad (4-4)$$

这个变换给出了直方图均衡化 w 的值，显然有

$$\int_0^s p_s(y) dy = w \quad (4-5)$$

该式实际上给出了另一个变换函数 $w = T_2(s)$ ，这正是我们希望得到的结果。因此，上述处理过程可分为三步：

- 1) 对给定图像执行直方图均衡化。
- 2) 根据希望得到的直方图求得变换函数 $w = T_2(s)$ 。
- 3) 对 $w = T_2(s)$ 求得反变换，并把第1步直方图均衡化的结果代入反变换。

图4-3a和图4-3b显示了对图4-2a的图像执行上述方法的结果。这个直方图由 $p_s(s) = \alpha e^{-\alpha s}$ 产生，其中 α 为一正常数。结果是增强了低灰度级别值，同时减少了高灰度级别值。图4-3b显示了这一结果，但此时是离散的，只能看到一个趋势。而图4-3d执行了随机增量式直方图夸张化方法，处理结果就很容易看出来。

4.11 为什么除了进行直方图均衡化之外还希望实现其他的操作

或许，为了实现某些特定的效果，我们需要突出图像的某些灰度值。由于人眼对图像中不同亮度的观察是有差异的，为了弥补这一差异，我们需要突出图像的某些灰度值。这也是我们执行直方图夸张化的原因：直方图夸张化能产生更满意的图像。

• 例4.1 给定一幅图像，其概率密度函数为

$$p_r(r) = Ae^{-r}$$

其中， r 为从0到 b 变化的灰度级变量， A 为归一化因子。计算变换 $s = T(r)$ ，其中 s 为变换图像的灰度级，使得变换图像的概率密度函数为

130

$$p_s(s) = Bse^{-s^2}$$

其中， s 的变化范围为0到 b ， B 为某一归一化因子。

用式(4-1)可以算出 $s = T(r)$

$$Bse^{-s^2} ds = Ae^{-r} dr$$

对等式两边积分，可以得到关于变量 s 和 r 的分布函数的等式。为了避免混淆，我们在左边用 y 代替积分变量，右边用 x 代替积分变量：

$$B \int_0^s ye^{-y^2} dy = A \int_0^r e^{-x} dx \quad (4-6)$$

式(4-6)左边为

$$\int_0^s ye^{-y^2} dy = \frac{1}{2} \int_0^s e^{-y^2} dy^2 = -\frac{1}{2} e^{-y^2} \Big|_0^s = \frac{1 - e^{-s^2}}{2} \quad (4-7)$$

式(4-6)右边为

$$\int_0^r e^{-x} dx = -e^{-x} \Big|_0^r = 1 - e^{-r} \quad (4-8)$$

把式(4-7)和式(4-8)代入式(4-6)有:

$$\begin{aligned}\frac{1 - e^{-s^2}}{2} &= \frac{A}{B}(1 - e^{-r}) \Rightarrow \\ e^{-s^2} &= 1 - \frac{2A}{B}(1 - e^{-r}) \Rightarrow \\ -s^2 &= \ln \left[1 - \frac{2A}{B}(1 - e^{-r}) \right] \Rightarrow \\ s &= \sqrt{-\ln \left[1 - \frac{2A}{B}(1 - e^{-r}) \right]}\end{aligned}$$

•
•
•

4.12 如果图像的对比度不均—该如何处理

上述讨论的图像增强直方图处理方法是全局性的,也就是说,直方图是基于整幅图像像素的变换而修改的。但是,一幅图像的不同部分可能有不同的图像质量。例如,一幅图像仅在图像中间部分有一很宽的阴影带,而图像的顶部和底部都很清晰。这种情况下,解决的办法就是用上述技术对图像执行局部处理:用一个邻域窗口去扫描图像,直方图局部处理仅在这个窗口中进行,即仅在此窗口内修改图像像素的灰度值。现在已有许多有效的局部处理的算法出现。

131

图4-4a显示了一幅需要执行局部图像增强的经典图像。这幅图像是在室内向着窗口拍摄的,因此图像中窗外那部分的质量是好的,但室内则明显曝光不够。图4-4b显示了对图4-4a执行全局直方图均衡化后的图像,我们虽然能看到室内的细节部分,但却把窗外部分过度曝光了,显然这个结果不能让我们满意。图4-4c显示了执行局部直方图均衡化后的图像,这幅图像的整体亮度看起来要均衡很多。这里我们用的窗口的大小是 40×40 ,而原图的大小为 400×400 。此时图像给我们的印象是没有任何部分是曝光不足或曝光过度的。但事实上,也有一小部分图像看起来是损坏了:在图像的底部和图像顶部的一小部分。这是因为原图中这一部分的光线严重不足,本来就没有记录太多的细节,于是在这个图像增强的过程中,我们同时也把斑点和噪音增强了。这个效果在图4-5中表现得更加明显。局部直方图均衡化(结果为图4-5c)的目的是改善图像看起来黑黑的那部分,以便能观察到更多的细节。但是,图像增强使得图像的这部分增加了原本没有的图像内容。

图4-6c显示了用图4-6a执行局部直方图增强产生的结果,这个结果与上述用局部直方图处理能得到更满意的结果不一样。图4-6a是一幅在捷克共和国的Karlstejn城堡前拍摄的图像,图像背景部分是城堡平坦的灰色墙。我们增强图像左边黑色区域时发现,采用局部直方图均衡化扩大了墙每一部分的微小变化,使得墙看起来跟粗糙的岩石外表差不多。尽管我们处理时以为采用局部直方图均衡化会产生更好的结果,但事实上,采用全局直方图均衡化产生的图像要好得多。

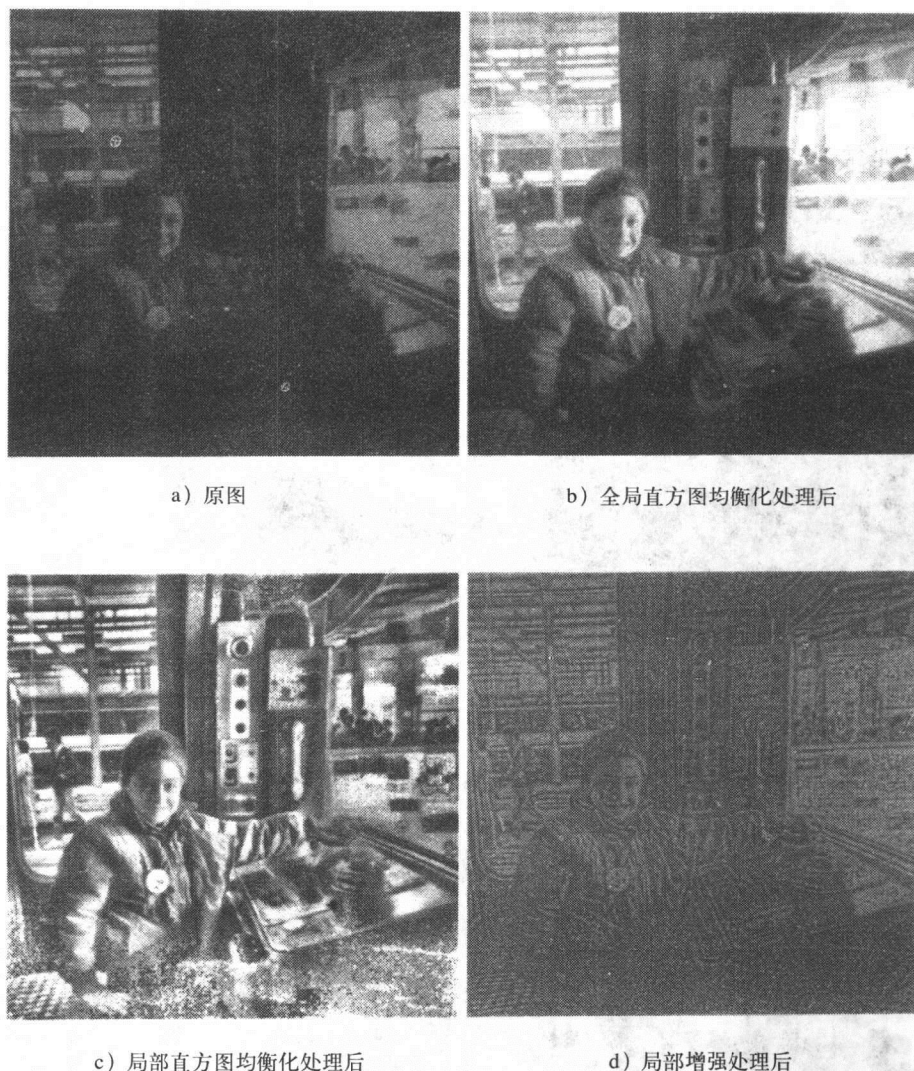


图4-4 年轻驾驶员图像的增强

4.13 对直方图的操作是否还有其他的方法

回答是肯定的，我们可以选取分布在窗口内的像素的均值和标准差。记以像素点 (x, y) 为中心的窗口内像素的灰度值均值为 $m(x, y)$ ，窗口内像素的方差为 $\sigma(x, y)$ ，像素点 (x, y) 的灰度值为 $f(x, y)$ 。于是每一个窗口内方差的增强变换为：

$$g(x, y) = A[f(x, y) - m(x, y)] + m(x, y) \quad (4-9)$$

其中 A 是比例因子。

我们希望尽可能地增强低灰度变量, 所以选取与 $\sigma(x, y)$ 成反比的增因子 A :

$$A = \frac{kM}{\sigma(x, y)}$$

其中 K 是常量, M 是整幅图像的平均灰度值。

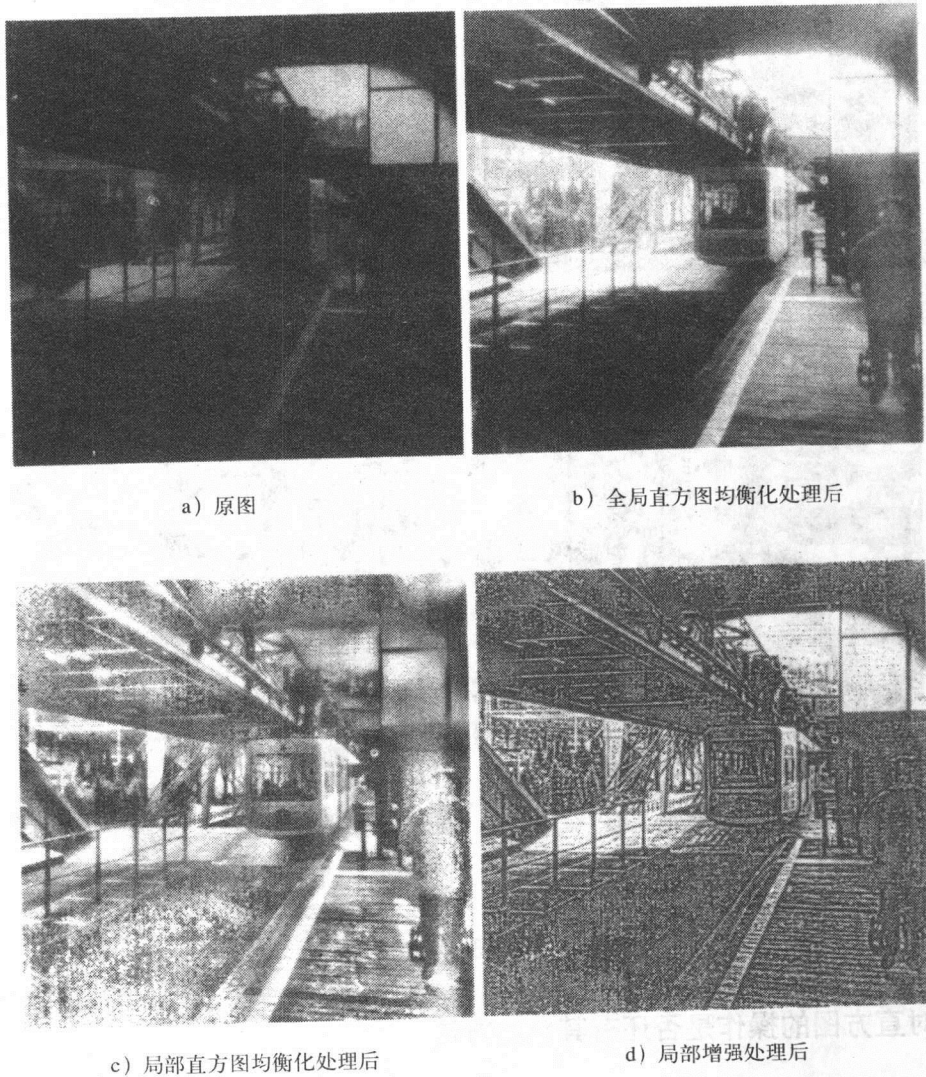
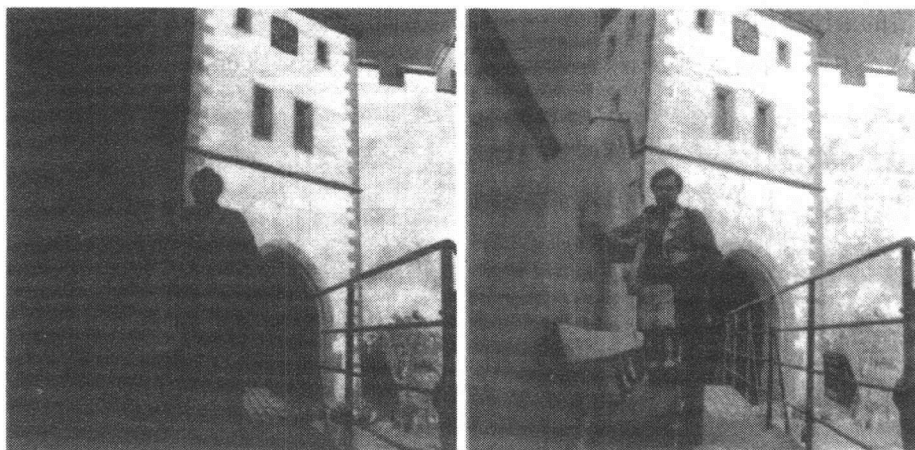


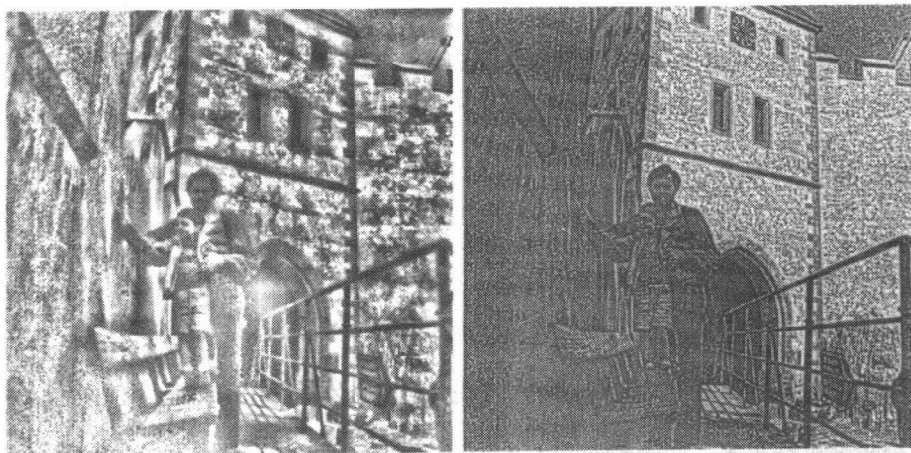
图4-5 伍珀塔尔 (Wuppertal) 磁悬浮列车图像的增强

图4-4d显示了对图4-4a中的图像执行 $k=3$ 和窗口大小为 5×5 的处理的结果。注意到, 此时虽然图像的细节很清楚, 但是整幅图像太黑, 这不能让人满意。图4-5d和图4-6d分别显示了对图4-5a和图4-6a中的图像执行增量式直方图均衡化处理的结果。



a) 原图

b) 全局直方图均衡化处理后



c) 局部直方图均衡化处理后

d) 局部增强处理后

图4-6 Karlstejn城堡图像的增强

4.14 如何提高多光谱图像的对比度

一幅多频带图像 (multiband image)、多光谱图像 (multispectral image) 或彩色图像由几个相同尺寸的图像分量构成, 每一图像分量对应一频谱分量。每一图像分量都是带有特定频谱的灰度图像, 其灰度值表示该像素在这个频带上的亮度。一个我们熟悉的例子就是有三个频带的红(R)、绿(G)、蓝(B)彩色图像。每幅图像都有三个频带, 对应三幅灰度图像分量。换句话说, 彩色图像上的每个像素都有三个分量值, 用三维空间的向量表示。其中, 每个频带对应三维空间的一个坐标分量。显然, 彩色图像的像素在彩色坐标空间形成点簇。

如果我们只拿其中一个频谱分量来进行处理, 那么我们将选择那个带有最多细节、拥有最大对比度且灰度值分布最广的频谱分量。

像素值的最佳延伸方向可能并不是某个坐标轴方向,而是另外一个方向(如图4-7a所示)。为了识别这个方向,我们就要对图像进行主分量分析(principal component analysis)即执行K-L变换。

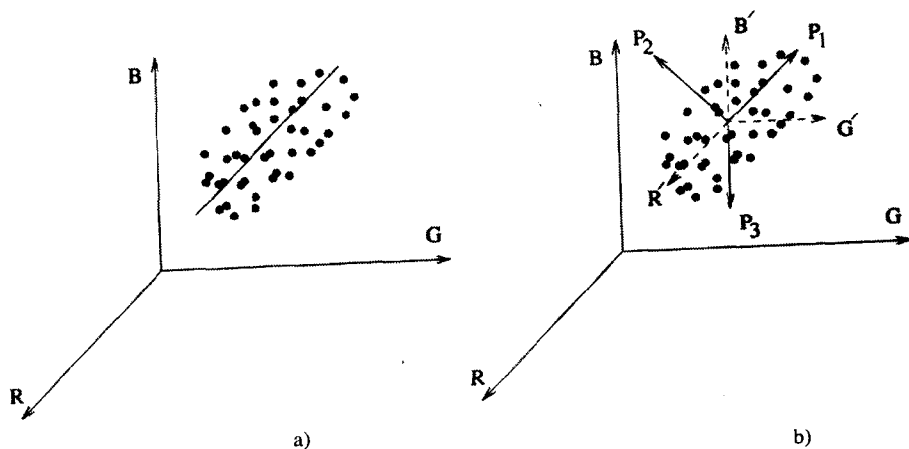


图4-7 彩色图像的像素在彩色坐标空间形成点簇。

点簇的最佳延伸方向并不平行于任一坐标轴

4.15 什么是主分量分析

主分量分析(即K-L变换)是一个坐标系的线性变换。变换使得新坐标系三个轴的方向分别与像素点簇分布的三个最大延伸方向一致。新坐标系中不同坐标轴上的点是不相关的,这就是说,由像素的三个向量分量所构成的三个灰度图像是完全不相关的。

4.16 本章讨论的K-L变换和第3章所讨论的有什么联系

通过分析随机结果集,两者都能产生不相关的图像分量。但是,在第3章中,整幅图像被看成一个随机试验的结果,而无其他随机结果可用。其他结果不可用性可由随机遍历性假定补偿。所以,尽管整体统计是基于单幅图像进行空间统计的,但他假定为是基于所有图像进行平均计算所得到的。本章中,单个像素值是一个随机试验的结果,而且我们对图像所有像素进行同样的处理。

4.17 如何实现主分量分析

为了实现主分量分析,必须先把数据的协方差矩阵对角化。随机试验的自协方差函数为:

$$C(i, j) \equiv E\{(x_i(k, l) - x_{i0})(x_j(k, l) - x_{j0})\}$$

其中 $x_i(k, l)$ 为像素点 (k, l) 在 i 频带的值, x_{i0} 为 i 频带的平均值, $x_j(k, l)$ 为像素点 (k, l) 在 j 频带的值, x_{j0} 为 j 频带的平均值。上述公式也可表示为:

$$C(i, j) = \frac{1}{N^2} \sum_{k=1}^N \sum_{l=1}^N (x_i(k, l) - x_{i0})(x_j(k, l) - x_{j0}) \quad (4-10)$$

因为图像有R、G、B三个频带，所以变量*i*和*j*只有三种取值，协方差矩阵为 3×3 矩阵。又由于数据不相关，所以*C*是一个对角化的矩阵，即当*i* ≠ *j*时有 $C(i, j) = 0$ 。为了达到这个目的，必须用由原数据的协方差矩阵的特征向量构成的变换矩阵对数据进行变换。处理步骤如下：

- 1) 在彩色空间中找到分布点的均值，记为(R_0, G_0, B_0)。
- 2) 在每个相应的频带上减去平均灰度值。这等价于把RGB坐标系的中心移至点分布的中心（见图4-7b中的坐标轴*R'G'B'*）。
- 3) 算出原分布的自相关矩阵 $C(i, j)$ （其中*i*和*j*的取值范围为R、G、B）。
- 4) 找出矩阵 $C(i, j)$ 的特征值并降序排列。用相应的特征向量作为行，得到特征向量矩阵*A*。

137

- 5) 用矩阵*A*变换原分布。用 $y = Ax$ 把每个三维向量 $x = \begin{pmatrix} R \\ G \\ B \end{pmatrix}$ 变换为 $y = \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix}$ ，即

$$y_k = \sum_i a_{ki} x_i。$$

这是一个线性变换。新的“颜色”是原来三种颜色亮度值的线性组合，所以第一个主分量包含图像的大部分信息（见图4-7b）。

4.18 使用主分量来表示图像的优点是什么

使用主分量来表示图像的优点是：

- 1) 使用一定的比特数，每一频带传输的信息是最大的，因为频带是不相关的，而且一个频带的信息不能通过其他频带预测得到。
- 2) 如果采用单色图像，可以只取第一主分量，这样不但能保证图像有最大的对比度，而且能保证用单个频带能传输最可能多的信息。

图4-8显示了一个主分量分析的例子。虽然初看起来，图4-8a、4-8b、4-8c和4-8d好像没有什么大的差别，但通过认真观察，我们会发现第一主分量图像结合了R、G、B三个频带的最好特征：比如，图4-8b和图4-8c中男孩的面的对比度要比图4-8a中的要高；图4-8a和图4-8b中男孩的右腿的对比度要比图4-8c中的要高；而在图4-8d中，上述两个地方都有很好的对比度。类似地，在图4-8a和图4-8b中，裤子和地面不存在对比，而图4-8c中有明显地对比，这个对比在图4-8d中同样有显示。

4.19 主分量分析的缺点是什么

由主分量分析构成的频带，其上的灰度值并没有物理意义，它们并不是任何自然的颜色。因此，不能简单地用像素的灰度值来对像素进行分类。这个性质在根据像素灰度值进行分类的遥感应用中，表现得相当准确。例如，在一个主分量频带中，代表水的像素可能会比图像中代表其他内容的像素更暗或者更亮，这个性质在不同的光谱频带中具有一致性。遥感科学家知道这个性质，所以用它来做识别。

138

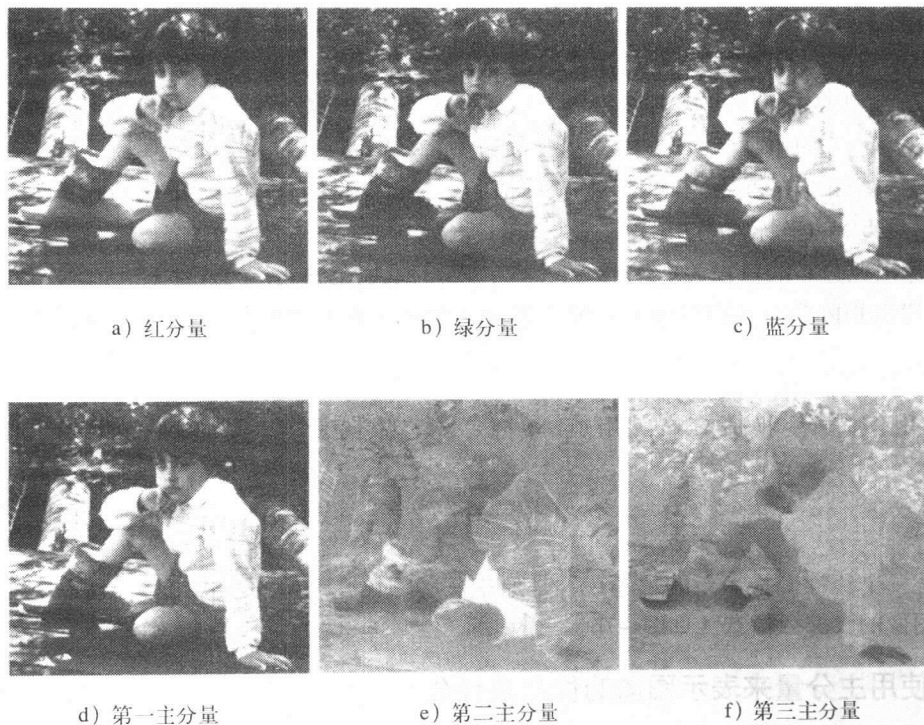


图4-8 一幅彩色图像的主分量分析

● 例4.2 下述矩阵 C 是某一具有三频带的图像的自协方差矩阵吗？

$$C = \begin{pmatrix} -1 & 0 & 1 \\ 0 & 1 & -2 \\ -2 & 2 & 0 \end{pmatrix}$$

根据式(4-10)，可知 C 肯定是对角元素为正的对称矩阵，所以上述矩阵 C 不可能是某一图像的自协方差矩阵。

例4.3 设一个三频带的图像的频带分别是绿、蓝和红，他们各自的灰度值均值分别为3、2和3，其自协方差矩阵为：

$$B = \begin{pmatrix} 2 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 2 \end{pmatrix}$$

又设一个像素在这绿、蓝和红三频带中的灰度值分别为5、3和4，那么这个像素在图像执行主分量分析后的三个主分量频带上的值分别是多少？

首先我们计算矩阵 B 的特征值:

$$\begin{vmatrix} 2-\lambda & 0 & 1 \\ 0 & 2-\lambda & 0 \\ 1 & 0 & 2-\lambda \end{vmatrix} = 0$$

$$\Rightarrow (2-\lambda)^3 - (2-\lambda) = 0 \Rightarrow (2-\lambda) \left[(2-\lambda)^2 - 1 \right] = 0$$

$$\Rightarrow (2-\lambda)(1-\lambda)(3-\lambda) = 0$$

因此, $\lambda_1 = 3$, $\lambda_2 = 2$, $\lambda_3 = 1$ 。于是相应的特征值为:

$$\begin{pmatrix} 2 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = 3 \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \Rightarrow \begin{array}{l} 2x_1 + x_3 = 3x_1 \\ 2x_2 = 3x_2 \\ x_1 + 2x_3 = 3x_3 \end{array} \left| \Rightarrow \begin{array}{l} x_1 = x_3 \\ x_2 = 0 \end{array} \right| \Rightarrow$$

$$\Rightarrow \mathbf{u}_1 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

$$\begin{pmatrix} 2 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = 2 \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \Rightarrow \begin{array}{l} 2x_1 + x_3 = 2x_1 \\ 2x_2 = 2x_2 \\ x_1 + 2x_3 = 2x_3 \end{array} \left| \Rightarrow \begin{array}{l} x_2 \text{ 任何值} \\ x_1 = x_3 = 0 \end{array} \right| \Rightarrow$$

$$\Rightarrow \mathbf{u}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \Rightarrow \begin{array}{l} 2x_1 + x_3 = x_1 \\ 2x_2 = x_2 \\ x_1 + 2x_3 = x_3 \end{array} \left| \Rightarrow \begin{array}{l} x_1 = -x_3 \\ x_2 = 0 \end{array} \right| \Rightarrow$$

$$\Rightarrow \mathbf{u}_3 = \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

140

于是, 变换矩阵 A 为:

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{pmatrix}$$

于是, 像素 $\begin{pmatrix} 5 \\ 3 \\ 4 \end{pmatrix}$ 变换为:

$$\begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 5 \\ 3 \\ 4 \end{pmatrix} = \begin{pmatrix} \frac{9}{\sqrt{2}} \\ 3 \\ -\frac{1}{\sqrt{2}} \end{pmatrix}$$

例4.4 一个 4×4 三频带的图像如下所示:

$$R = \begin{pmatrix} 3 & 3 & 5 & 6 \\ 3 & 4 & 4 & 5 \\ 4 & 5 & 5 & 6 \\ 4 & 5 & 5 & 6 \end{pmatrix} \quad G = \begin{pmatrix} 3 & 2 & 3 & 4 \\ 1 & 5 & 3 & 6 \\ 4 & 5 & 3 & 6 \\ 2 & 4 & 4 & 5 \end{pmatrix} \quad B = \begin{pmatrix} 4 & 2 & 3 & 4 \\ 1 & 4 & 2 & 4 \\ 4 & 3 & 3 & 5 \\ 2 & 3 & 5 & 5 \end{pmatrix}$$

计算图像的三个主分量并证明三个分量不相关。

首先我们计算每一频带的均值:

$$\begin{aligned} R_0 &= \frac{1}{16}(3+3+5+6+3+4+4+5+4+5+5+6+4+5+5+6) \\ &= \frac{73}{16} = 4.5625 \\ G_0 &= \frac{1}{16}(3+2+3+4+1+5+3+6+4+5+3+6+2+4+4+5) \\ &= \frac{60}{16} = 3.75 \\ B_0 &= \frac{1}{16}(4+2+3+4+1+4+2+4+4+3+3+5+2+3+5+5) \\ &= \frac{50}{16} = 3.125 \end{aligned}$$

[141]

接着我们计算协方差矩阵的元素:

$$\begin{aligned} C_{RR} &= \frac{1}{16} \sum_{k=1}^4 \sum_{l=1}^4 (R(k,l) - R_0)^2 = 0.996094 \\ C_{RG} &= \frac{1}{16} \sum_{k=1}^4 \sum_{l=1}^4 (R(k,l) - R_0)(G(k,l) - G_0) = 0.953125 \\ C_{RB} &= \frac{1}{16} \sum_{k=1}^4 \sum_{l=1}^4 (R(k,l) - R_0)(B(k,l) - B_0) = 0.726563 \\ C_{GG} &= \frac{1}{16} \sum_{k=1}^4 \sum_{l=1}^4 (G(k,l) - G_0)^2 = 1.9375 \\ C_{GB} &= \frac{1}{16} \sum_{k=1}^4 \sum_{l=1}^4 (G(k,l) - G_0)(B(k,l) - B_0) = 1.28125 \\ C_{BB} &= \frac{1}{16} \sum_{k=1}^4 \sum_{l=1}^4 (B(k,l) - B_0)^2 = 1.359375 \end{aligned}$$

因此, 协方差矩阵为:

$$C = \begin{pmatrix} 0.996094 & 0.953125 & 0.726563 \\ 0.953125 & 1.937500 & 1.281250 \\ 0.726563 & 1.28125 & 1.359375 \end{pmatrix}$$

这个矩阵的特征值为:

$$\lambda_1 = 3.528765 \quad \lambda_2 = 0.435504 \quad \lambda_3 = 0.328700$$

相应的特征向量为:

$$\mathbf{u}_1 = \begin{pmatrix} 0.427670 \\ 0.708330 \\ 0.561576 \end{pmatrix} \quad \mathbf{u}_2 = \begin{pmatrix} 0.876742 \\ -0.173808 \\ -0.448457 \end{pmatrix} \quad \mathbf{u}_3 = \begin{pmatrix} 0.220050 \\ -0.684149 \\ 0.695355 \end{pmatrix}$$

因此, 变换矩阵为:

$$A = \begin{pmatrix} 0.427670 & 0.708330 & 0.561576 \\ 0.876742 & -0.173808 & -0.448457 \\ 0.220050 & -0.684149 & 0.695355 \end{pmatrix}$$

用该矩阵对每一个像素值进行变换, 可得图像的主分量。例如, 我们对图像前几个像素的变换为:

$$\begin{pmatrix} 5.654302 \\ 0.314974 \\ 1.389125 \end{pmatrix} = \begin{pmatrix} 0.427670 & 0.708330 & 0.561576 \\ 0.876742 & -0.173808 & -0.448457 \\ 0.220050 & -0.684149 & 0.695355 \end{pmatrix} \begin{pmatrix} 3 \\ 3 \\ 4 \end{pmatrix}$$

142

$$\begin{pmatrix} 3.822820 \\ 1.385694 \\ 0.682562 \end{pmatrix} = \begin{pmatrix} 0.427670 & 0.708330 & 0.561576 \\ 0.876742 & -0.173808 & -0.448457 \\ 0.220050 & -0.684149 & 0.695355 \end{pmatrix} \begin{pmatrix} 3 \\ 2 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} 5.948065 \\ 2.516912 \\ 1.133874 \end{pmatrix} = \begin{pmatrix} 0.427670 & 0.708330 & 0.561576 \\ 0.876742 & -0.173808 & -0.448457 \\ 0.220050 & -0.684149 & 0.695355 \end{pmatrix} \begin{pmatrix} 5 \\ 3 \\ 3 \end{pmatrix}$$

变换后, 用三维向量表示的像素的第一个元素组成了原图像的第一主分量, 第二个元素组成了原图像的第二主分量, 同样, 第三个元素组成了原图像的第三主分量。因此, 我们有:

$$P_1 = \begin{pmatrix} 5.654302 & 3.822820 & 5.948065 & 7.645640 \\ 2.552915 & 7.498631 & 4.958820 & 8.634631 \\ 6.790301 & 7.364725 & 5.948065 & 9.623876 \\ 4.250490 & 6.656395 & 7.779546 & 8.915546 \end{pmatrix}$$

$$P_2 = \begin{pmatrix} 0.314974 & 1.385694 & 2.516912 & 2.771389 \\ 2.007960 & 0.844099 & 2.088630 & 1.547035 \\ 1.017905 & 2.169300 & 2.516912 & 1.975317 \\ 2.262436 & 2.343106 & 1.446188 & 2.149123 \end{pmatrix}$$

$$P_3 = \begin{pmatrix} 1.389125 & 0.682562 & 1.133874 & 1.365131 \\ 0.671360 & 0.240878 & 0.218468 & -0.223219 \\ 0.925027 & -0.234424 & 1.133874 & 0.692187 \\ 0.902617 & 0.449725 & 1.840433 & 1.376336 \end{pmatrix}$$

为了证明三个新频带上的数据不相关, 应该计算新数据的自协方差矩阵。首先, 我们计算出每个频带的均值: P_{10} 、 P_{20} 、 P_{30} , 接着我们计算:

$$\begin{aligned}
C_{P_1 P_1} &= \frac{1}{16} \sum_{i=1}^4 \sum_{j=1}^4 (P_1(i, j) - P_{10})^2 = 3.528765 \\
C_{P_1 P_2} &= \frac{1}{16} \sum_{i=1}^4 \sum_{j=1}^4 (P_1(i, j) - P_{10})(P_2(i, j) - P_{20}) = 0.0 \\
C_{P_1 P_3} &= \frac{1}{16} \sum_{i=1}^4 \sum_{j=1}^4 (P_1(i, j) - P_{10})(P_3(i, j) - P_{30}) = 0.0 \\
C_{P_2 P_2} &= \frac{1}{16} \sum_{i=1}^4 \sum_{j=1}^4 (P_2(i, j) - P_{20})^2 = 0.435504 \\
C_{P_2 P_3} &= \frac{1}{16} \sum_{i=1}^4 \sum_{j=1}^4 (P_2(i, j) - P_{20})(P_3(i, j) - P_{30}) = 0.0 \\
C_{P_3 P_3} &= \frac{1}{16} \sum_{i=1}^4 \sum_{j=1}^4 (P_3(i, j) - P_{30})^2 = 0.328700
\end{aligned}$$

我们可以看到，这个协方差矩阵是对角化的，所以三个新频带上的数据是不相关的。

例4.5 证明例4.4中图像的第一主分量比任何一个原频带有更好的对比度。

一幅图像的对比度可用灰度值的取值范围宽度来描述。原图中，红频带的取值范围宽度为3，绿频带为5，蓝频带为4。而在第一主分量中，灰度值的取值范围宽度为：9.623876 - 2.552915 = 7.070961。显然，第一主分量的宽度比先前的要大。

4.20 有一些增强了对比度的图像看起来噪声很明显。如何处理这种情况

的确，如图4-4c、图4-5c和图4-6c显示，在执行局部直方图均衡化后，那些具有相同颜色的大块区域的窗口的噪声得到加强。因为这些像素的灰度值取值范围延伸到0~255整个取值范围，所以噪声也相应地增强了。因此我们必须对图像进行再处理，对一些噪声降噪。降噪声的方法取决于给定图像中噪声的类型。

4.21 图像噪声的类型有哪些

噪声有很多种。但是，它们可以分为两大类：加性噪声（additive noise）和乘性噪声（multiplicative noise）。光照变化就是一个乘性噪声的例子。乘性噪声也许是图像中最普通的噪声。加性噪声通常表现为脉冲噪声（impulse noise）或高斯噪声（Gaussian noise）。图4-9a显示了一幅被脉冲噪声污染的图像，图4-9b显示了一幅被零均值的加性高斯噪声污染的图像。

脉冲噪声随机改变一些像素值。在二值图像中表现为使一些像素点变白，使一些像素点变黑。这就是为什么脉冲噪声被称为是椒盐噪声（salt and pepper noise）。零均值的加性高斯

噪声是指将一个由零均值的高斯概率密度函数刻画的噪声加到每个像素中。

144

我们用排序滤波器去除脉冲噪声，用平滑降低高斯噪声。



a) 被脉冲噪声污染的图



b) 被加性高斯噪声污染的图



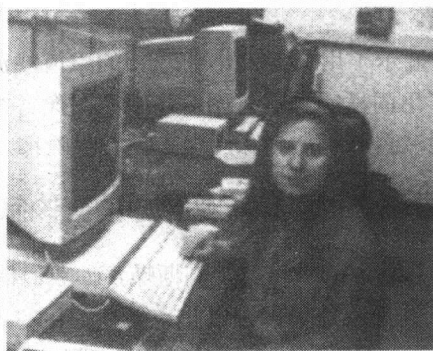
c) 用中值滤波器对图a) 滤波的结果



d) 用中值滤波器对图b) 滤波的结果



e) 用均值滤波器平滑图a) 的结果



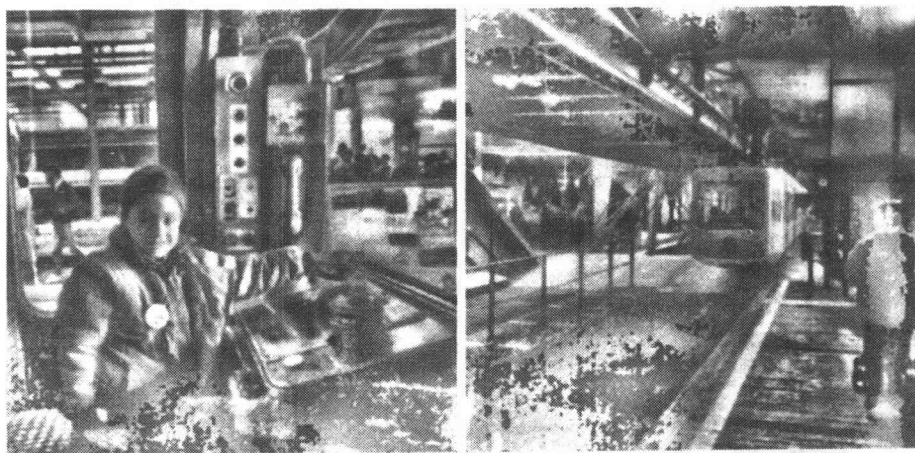
f) 用均值滤波器平滑图b) 的结果

图4-9 用滤波器降噪的例子

4.22 什么是排序滤波器

排序滤波器是指其输出值依赖于根据滤波器窗口内像素灰度值的排序顺序的滤波器。最常见的例子就是中值滤波器 (median filter)。

图4-10显示了用中值滤波器对图4-4c和图4-5c中的图像进行降噪的结果。



a) 用中值滤波器增强图4-4 c

b) 图4-5 c中的图像 (窗口大小为 3×3)

图 4-10

4.23 什么是中值滤波

一个数值集合的中值是指该集合中能把这个集合等分成两部分的一个元素，它使得其中一部分的值全部小于等于中值，而另外一部分的值全部大于或等于中值。例如，在一个 5×5 的窗口中，一共有25个灰度值，把这些灰度值按升序排序，那么中值就是序列中的第13个值。这样，中值滤波器使得其邻域拥有不同灰度值的点更接近于它的邻域点，因此它能消除一些孤立像素点。

图4-9c显示了用窗口大小为 5×5 的中值滤波器对图4-9a中的图像进行处理的结果，而图4-9d显示了用同样的方法对图4-9b中的图像（被高斯噪声污染的图像）进行处理的结果。显然，中值滤波器几乎能把脉冲噪声完全去除掉。

4.24 如果图像的噪声不是脉冲的应该怎么办

图像中最普通的噪声就是高斯噪声。我们能平滑图像达到去除高斯噪声的目的。例如，图像在各点的像素值可以用以该点为中心的平滑滤波窗口内的像素平均值代替。图4-9e和图4-9f分别显示了对图4-9a和图4-9b执行平滑滤波的结果，平滑滤波窗口跟中值滤波窗口大小相同，都为 5×5 。我们注意到，这种平滑滤波对降低高斯噪声十分有效，而对于脉冲噪声则没有那么好的效果。平滑滤波实际上是一种简单的低通滤波(lowpass filtering)。

4.25 为什么低通滤波可以减少噪声

通常, 添加在图像中的噪声是不相关的。也就是说, 噪声具有平直的频谱。然而, 大多数图像都有一个频谱范围, 它低频上的值较高, 高频上的值较小, 并且值随频率递增而递减。在达到一定频率后, 噪声信号的频谱值就会大于静图像信号的频谱值 (如图4-11所示)。如果我们使用低通滤波器, 则可以截断高频, 保留低频, 从而去除了噪声占优的高频部分。当然, 我们采用低通滤波器的同时也把一些有用的图像高频信号去除了。事实上, 滤波结果很好, 但也带来了图像一定程度的失真。

处理的步骤如下:

1) 对图像进行傅里叶变换。

2) 傅里叶变换的结果乘上一个低通滤波函数, 这个函数可以“截断”所有的高频部分而不会更改一定频率以下的低频部分。在二维频率空间中, 这个理想的低通滤波器如图4-12所示。

3) 对步骤2的结果执行傅里叶反变换。

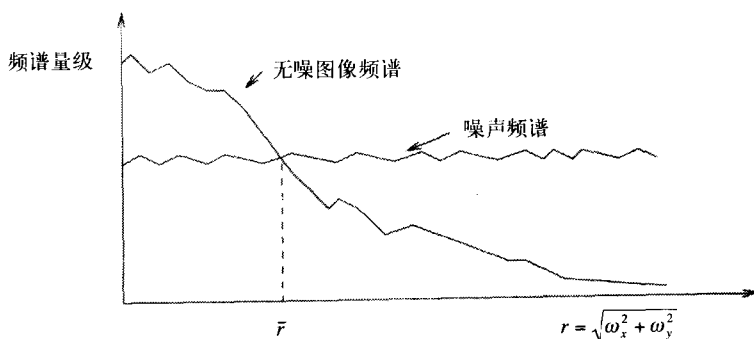


图4-11 达到一定频率后, 图像噪声的频谱占优。
低通滤波器的理想截断频率是 $r_0 = \bar{r}$

147

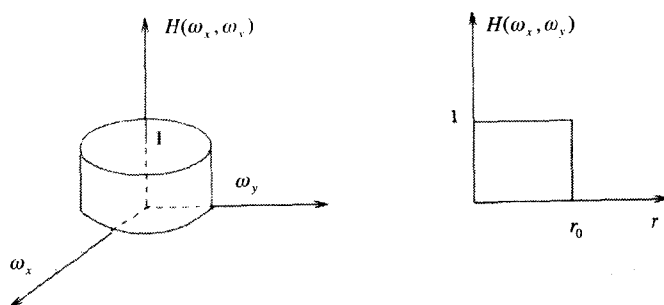


图4-12 二维频域中的理想低通滤波器, 右图为左图中的截断
频率为 $r_0 (r \equiv \sqrt{\omega_x^2 + \omega_y^2})$ 的理想低通滤波器的横截面

在频域中对两个频谱做乘法等价于在时域中对两个相应的函数做卷积。因此, 为了实现上述的低通滤波, 我们可在时域中寻找一个其傅里叶变换为理想低通滤波器的函数, 并且用

图像卷积这个函数。但这只是理想的结果。事实上,其傅里叶变换为理想低通滤波器的函数是不存在的,因为这个函数可以说是定义在无限区间上的。

4.26 如果我们感兴趣的是图像的高频部分应该怎么做

如果我们需要增强的是图像的一些细小细节,这个图像处理方法称为图像锐化(sharpening)。它增强的是图像灰度的细小变化,当然噪声也同时得到增强。

为了实现锐化,我们用数值差分公式计算每一像素点的局部亮度梯度。如果需要更复杂一点的话,我们可以采用前边讨论的平滑滤波方法。当然,实现这个处理的滤波器被称为高通滤波器,它允许保留高频成分并且截断低频成分。

4.27 什么是理想高通滤波器

频域中理想的高通滤波器如图4-13所示。

在频域中,用这样的高通滤波器滤波等价于在时域中卷积一个其傅里叶变换为这个高通滤波器的函数。但实际上,没有一个有限的函数对应于理想高通滤波器,所以人们常用各种近似来代替它。

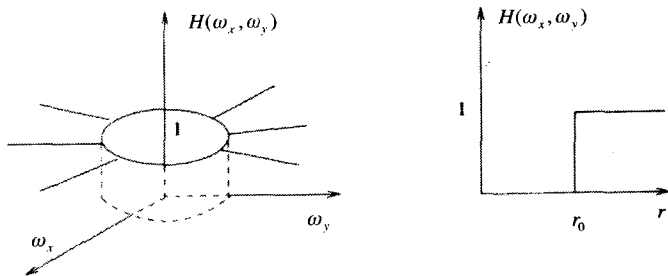


图4-13 理想高通滤波器的频谱在半径为 r_0 的圆内为零,在圆外为1,
右图为左图滤波器的横截面,其中 $r = \sqrt{\omega_x^2 + \omega_y^2}$

4.28 如何改善有光照变化的图像

如果我们知道每个表示图像的函数 $f(x, y)$ 能被表达成其两个分量的乘积的话,就可以改善有光照变化的图像。函数 $f(x, y)$ 的两个分量为:光照函数 $i(x, y)$ 和反射函数 $r(x, y)$ 。其中, $r(x, y)$ 的性质取决于成像物体的表面特性。

$$f(x, y) = i(x, y)r(x, y) \quad (4-11)$$

光照强度一般具有一致性,在图像的傅里叶变换下表现为低频分量。然而不同的材料或物体的反射率差异很大,常引起反射函数 $r(x, y)$ 急剧变化,从而使图像的灰度值发生突变,这种变化与高频分量有关。对式(4-11)两边取对数,可使图像函数 $f(x, y)$ 的两个分量分离,即把相乘的分量转化为相加的分量: $\ln f(x, y) = \ln i(x, y) + \ln r(x, y)$

接着,我们用一个叫同态滤波器(homomorphic filter)的滤波器来对这个取对数后的图像执行滤波。同态滤波器能减少低频并增加高频,从而能减少光照变化并锐化边缘或细节。

同态滤波器的横截面如图4-14所示。

图4-15a和图4-16a显示了两幅光照从左向右均匀变化的图像。而图4-15b和图4-16b显示了对这两幅图像执行同态滤波后的结果,改善的结果很乐观,不但大大减少了图像的光照变化,而且原图较暗部分的一些细节也变得清晰可见。上述结果是对图像取对数后,用如下变换函数来滤波的:

$$H(\omega_x, \omega_y) = \frac{1}{1 + e^{-s(\sqrt{\omega_x^2 + \omega_y^2} - \omega_0)}} + A \quad (149)$$

其中, $s = 1$, $\omega_0 = 128$, $A = 10$ 。这个同态滤波器的参数也与图4-14中的参数 γ_H 和 γ_L 相关:

$$\gamma_L = \frac{1}{1 + e^{s\omega_0}} + A, \quad \gamma_H = 1 + A$$

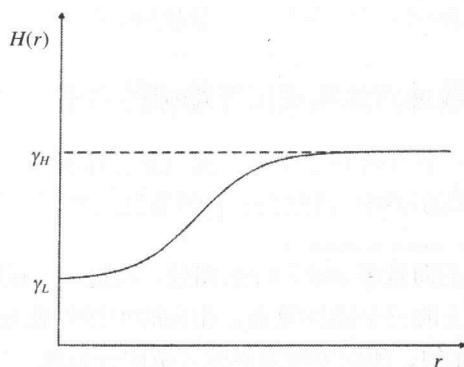


图4-14 一个同态滤波器函数的横截面, $r = \sqrt{\omega_x^2 + \omega_y^2}$



a) 原图

b) 执行同态滤波后的结果

图4-15 对图像执行同态滤波的例子



a) 原图

b) 执行同态滤波后的结果

图4-16 对图像执行同态滤波的例子

4.29 是否可以用第2章的线性方法实现图像增强的目的

回答是肯定的。通常，用一个合适的滤波器对图像进行卷积可以实现图像平滑和图像锐化。这就是我们喜欢有限滤波器的原因：图像上的有限卷积滤波可看成对图像执行线性运算。

- 例4.6 给定一幅可用一个 9×1 的向量表示的 3×3 的图像，构造一个矩阵，它能平滑图像的列，
- 平滑时使用所平滑点的列方向上的三个连续像素，相应的平滑权重分别为 $1/4$ 、 $1/2$ 、 $1/4$ 。假定图像每个方向都是周期性变化的，所以图像边界也可做同样处理。

记原图为：

$$\begin{pmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{pmatrix}$$

并且平滑后的图为：

$$\begin{pmatrix} \tilde{g}_{11} & \tilde{g}_{12} & \tilde{g}_{13} \\ \tilde{g}_{21} & \tilde{g}_{22} & \tilde{g}_{23} \\ \tilde{g}_{31} & \tilde{g}_{32} & \tilde{g}_{33} \end{pmatrix}$$

把平滑矩阵记为 A ，它的元素为 a_{ij} ：

$$\begin{pmatrix} \tilde{g}_{11} \\ \tilde{g}_{21} \\ \tilde{g}_{31} \\ \tilde{g}_{12} \\ \tilde{g}_{22} \\ \tilde{g}_{32} \\ \tilde{g}_{13} \\ \tilde{g}_{23} \\ \tilde{g}_{33} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{19} \\ a_{21} & a_{22} & \dots & a_{29} \\ a_{31} & a_{32} & \dots & a_{39} \\ a_{41} & a_{42} & \dots & a_{49} \\ a_{51} & a_{52} & \dots & a_{59} \\ a_{61} & a_{62} & \dots & a_{69} \\ a_{71} & a_{72} & \dots & a_{79} \\ a_{81} & a_{82} & \dots & a_{89} \\ a_{91} & a_{92} & \dots & a_{99} \end{pmatrix} \begin{pmatrix} g_{11} \\ g_{21} \\ g_{31} \\ g_{12} \\ g_{22} \\ g_{32} \\ g_{13} \\ g_{23} \\ g_{33} \end{pmatrix}$$

根据上面的等式, 可得:

$$\begin{aligned}\tilde{g}_{11} = & a_{11}g_{11} + a_{12}g_{21} + a_{13}g_{31} + a_{14}g_{12} + a_{15}g_{22} \\ & + a_{16}g_{32} + a_{17}g_{13} + a_{18}g_{23} + a_{19}g_{33}\end{aligned}\quad (4-12)$$

根据给定的平滑定义, 可得:

$$\tilde{g}_{11} = \frac{1}{4}g_{31} + \frac{1}{2}g_{11} + \frac{1}{4}g_{21}\quad (4-13)$$

比较式(4-12)和式(4-13), 可得:

$$a_{11} = \frac{1}{2}, a_{12} = \frac{1}{4}, a_{13} = \frac{1}{4}, a_{14} = a_{15} = \dots = a_{19} = 0$$

类似地, 矩阵其他元素也可算出, 因此这个矩阵可以表示如下:

$$A = \begin{pmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \end{pmatrix}$$

4.30 本章要点

本章我们主要讨论了图像增强。图像增强, 就是根据主观标准去处理图像, 使图像看起来更好。而大多数处理方法都是非线性的, 因此不能用前几章的理论进行描述。一些方法不仅是非线性的, 而且是非齐次的。

图像的对比度增强, 可以使图像的层次更丰富, 这种增强可以用改变图像像素灰度值的方法来实现。而且, 这种增强可以使处理后的直方图满足一定的形状。其中, 形状的选择是任意的。然而, 如果处理的图像是多频带的图像, 则可以在多维空间中(像素的灰度值在沿相应轴的不同频带上测量得到)定义一个方向, 当我们把数据投影到这个方向上的时候, 得到的投影值能达到最大(见图4-7a)。这个方向定义了多维空间中数据第一个主轴的方向(三像素值)。一个新坐标轴对应一个新频带, 把像素投影到这个第一轴时, 可以得到一幅灰度图像。这幅灰度图像叫做多频谱图像的第一主分量, 对所有的灰度频谱都可以由这幅多频带图像来构造。其中, 第一主分量拥有最大的对比度和最多的信息。但是, 因为新频带灰度值是图像原频带灰度值的线性组合, 所以该灰度值没有真正的物理意义。直方图处理和主向量分

析的最根本的不同在于，前者中，像素的值可以任意地改变；后者中，像素灰度值的联合概率密度函数保持不变，它只是用一个更方便的方式来“读取”数据而已，并没有引进原来没有的信息（例如，比较图4-6c和图4-8d增强的图像）。

同样，我们也可以在傅里叶频谱中增强图像：可以很好地去掉任意不需要的频带。这个方法可以由拥有预先定义的频谱的卷积滤波器来完成。因此，我们有必要先了解滤波器，弄明白每个不同的任务需要什么样的滤波器。这就是我们将在下一章中讨论的内容。

153
154

第5章 二维滤波器

5.1 本章概述

图像处理经常需要忽略或增强特定的空间频率细节信息。这相当于用特定的函数对图像进行傅里叶变换,以去掉特定的频率成分或对其进行修改。我们称这种做法为对图像进行滤波(filter the image),同时把所使用的特定函数称为滤波器。

本章探讨二维滤波器的一些基本性质并且介绍一些特殊的傅里叶变换,这些傅里叶变换可以转换成直接对图像做简单的卷积运算,这样可以避免使用傅里叶变换本身。

5.2 如何定义一个二维滤波器

二维滤波器定义为傅里叶变换的形式 $H(\mu, \nu)$,称为系统函数(system function)。对 $H(\mu, \nu)$ 进行傅里叶反变换可以获得实域上的滤波器。称它为滤波器的单位采样响应(unit sample response),记为 $h(k, l)$ 。

5.3 系统函数和滤波器的单位采样响应是如何联系在一起的

将 $H(\mu, \nu)$ 定义成 (μ, ν) 的连续函数。定义单位采样响应 $h(k, l)$ 是 $H(\mu, \nu)$ 的傅里叶反变换,但是因为 $h(k, l)$ 将用于数字图像的卷积,它只需定义在离散点上,因此两者的关系可表示为:

$$h(k, l) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} H(\mu, \nu) e^{j(\mu k + \nu l)} d\mu d\nu \quad (5-1)$$

$$H(\mu, \nu) = \frac{1}{2\pi} \sum_{n=-\infty}^{+\infty} \sum_{m=-\infty}^{+\infty} h(n, m) e^{-j(\mu n + \nu m)} \quad (5-2) \quad \boxed{155}$$

假如我们只关心那些输出值是实的滤波器,那么这些等式可以改为:

$$h(k, l) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} H(\mu, \nu) \cos(\mu k + \nu l) d\mu d\nu \quad (5-3)$$

$$H(\mu, \nu) = \frac{1}{2\pi} \sum_{n=-\infty}^{+\infty} \sum_{m=-\infty}^{+\infty} h(n, m) \cos(\mu n + \nu m) \quad (5-4)$$

5.4 为什么我们对实域的滤波器函数感兴趣

通过将图像与 $h(k, l)$ 做简单的卷积运算而不是与 $H(\mu, \nu)$ 进行傅里叶变换,可以达到对图像做所必要的增强。图5-1显示了一维的情形。对于二维来说是完全类似的。

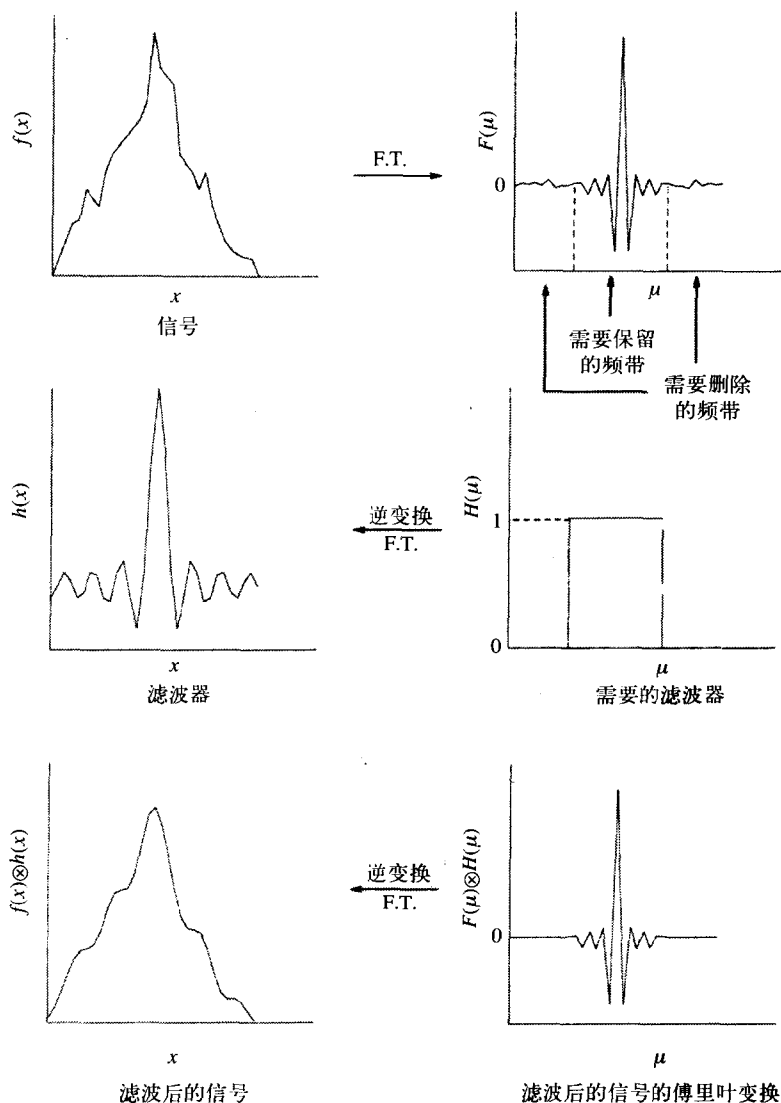


图5-1 顶行：分别是信号和它对应的傅里叶变换。中间行：左边是滤波器的单位采样响应函数，右边是滤波器的系统函数。底行：左边是顶行的信号和中间行滤波器卷积后得到的滤波后的信号。右边是滤波后的信号所对应的傅里叶变换，它是顶行的信号所对应的傅里叶变换与中间行的滤波器对应的傅里叶变换（系统函数）之积

156

5.5 $h(k, l)$ 需要满足什么条件才可以做为一个卷积滤波器

应当存在有限的 K 和 L ，当 $k > K$ 而且 $l > L$ 时 $h(k, l)$ 必须为零；即，对图像进行卷积的滤波器是个数有限的数列。因此理想低通、理想带通和理想高通滤波器都不能满足这个条件。

B5.1 理想低通滤波器的单位采样响应是怎样的?

对理想的低通滤波器来说, 当频率大于某个特定值 R 时其值为0, 定义为:

$$H(\mu, \nu) = \begin{cases} 1 & \text{当 } \sqrt{\mu^2 + \nu^2} \leq R \text{ 时} \\ 0 & \text{其他} \end{cases} \quad (5-5)$$

由式(5-3), 用该 $H(\mu, \nu)$ 的定义可以计算其相应的单位采样响应:

$$h(k, l) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \cos(\mu k + \nu l) H(\mu, \nu) d\mu d\nu$$

我们将频率空间坐标系 (μ, ν) 转换为极坐标系 (r, θ) :

$$\left. \begin{aligned} \mu &\equiv r \cos \theta \\ \nu &\equiv r \sin \theta \end{aligned} \right\} \Rightarrow \mu^2 + \nu^2 = r^2, \quad d\mu d\nu = r dr d\theta$$

因此

$$h(k, l) = \frac{1}{2\pi} \int_0^{2\pi} \int_0^R \cos(rk \cos \theta + rl \sin \theta) r dr d\theta \quad (5-6)$$

可写为

$$\begin{aligned} k \cos \theta + l \sin \theta &= \sqrt{k^2 + l^2} \left[\frac{k}{\sqrt{k^2 + l^2}} \cos \theta + \frac{l}{\sqrt{k^2 + l^2}} \sin \theta \right] \\ &= \sqrt{k^2 + l^2} [\sin \phi \cos \theta + \cos \phi \sin \theta] = \sqrt{k^2 + l^2} \sin(\theta + \phi) \end{aligned}$$

其中角度 ϕ 的定义使得

$$\sin \phi \equiv \frac{k}{\sqrt{k^2 + l^2}} \text{ 和 } \cos \phi \equiv \frac{l}{\sqrt{k^2 + l^2}}$$

定义新的变量 $t = \theta + \phi$ 。那么式(5-6)可以改写为:

$$\begin{aligned} h(k, l) &= \frac{1}{2\pi} \int_{\phi}^{2\pi+\phi} \int_0^R \cos(r\sqrt{k^2 + l^2} \sin t) r dr dt \\ &= \frac{1}{2\pi} \int_{\phi}^{2\pi} \int_0^R \cos(r\sqrt{k^2 + l^2} \sin t) r dr dt \\ &\quad + \frac{1}{2\pi} \int_{2\pi}^{2\pi+\phi} \int_0^R \cos(r\sqrt{k^2 + l^2} \sin t) r dr dt \end{aligned}$$

在上式的第二项中, 令 $\tilde{t} = t - 2\pi \Rightarrow t = \tilde{t} + 2\pi \Rightarrow \sin t = \sin \tilde{t}$ 。因此有:

$$\begin{aligned} h(k, l) &= \frac{1}{2\pi} \int_{\phi}^{2\pi} \int_0^R \cos(r\sqrt{k^2 + l^2} \sin t) r dr dt \\ &\quad + \frac{1}{2\pi} \int_0^{\phi} \int_0^R \cos(r\sqrt{k^2 + l^2} \sin \tilde{t}) r dr d\tilde{t} \\ &= \frac{1}{2\pi} \int_0^{2\pi} \int_0^R \cos(r\sqrt{k^2 + l^2} \sin t) r dr dt \end{aligned}$$

该式可以改写为:

$$\begin{aligned} h(k, l) &= \frac{1}{2\pi} \int_0^\pi \int_0^R \cos(r\sqrt{k^2 + l^2} \sin t) r dr dt \\ &\quad + \frac{1}{2\pi} \int_\pi^{2\pi} \int_0^R \cos(r\sqrt{k^2 + l^2} \sin t) r dr dt \end{aligned}$$

158

对上式的第二项, 定义新的积分变量 $\tilde{t} = t - \pi \Rightarrow t = \tilde{t} + \pi \Rightarrow \sin t = -\sin \tilde{t} \Rightarrow \cos(r\sqrt{k^2 + l^2} \sin t) = \cos(r\sqrt{k^2 + l^2} \sin \tilde{t})$ 和 $dt = d\tilde{t}$, 则:

$$\begin{aligned} h(k, l) &= \frac{1}{2\pi} \int_0^\pi \int_0^R \cos(r\sqrt{k^2 + l^2} \sin t) r dr dt \\ &\quad + \frac{1}{2\pi} \int_0^\pi \int_0^R \cos(r\sqrt{k^2 + l^2} \sin \tilde{t}) r dr d\tilde{t} \\ &= \frac{1}{\pi} \int_0^R \left\{ \int_0^\pi \cos(r\sqrt{k^2 + l^2} \sin t) dt \right\} r dr \end{aligned} \quad (5-7)$$

我们知道零阶第一类Bessel函数可定义为:

$$J_0(z) \equiv \frac{1}{\pi} \int_0^\pi \cos(z \sin \theta) d\theta \quad (5-8)$$

假如将式(5-8)代入式(5-7), 得到:

$$h(k, l) = \int_0^R r J_0(r\sqrt{k^2 + l^2}) dr$$

定义新的积分变量 $x = r\sqrt{k^2 + l^2} \Rightarrow dr = \frac{1}{\sqrt{k^2 + l^2}} dx$, 那么有:

$$h(k, l) = \frac{1}{k^2 + l^2} \int_0^{R\sqrt{k^2 + l^2}} x J_0(x) dx \quad (5-9)$$

由Bessel函数理论, 可以知道:

$$\int x^{p+1} J_p(x) dx = x^{p+1} J_{p+1}(x) \quad (5-10)$$

在式(5-10)中取 $p = 0$, 代入式(5-9):

$$\begin{aligned} h(k, l) &= \frac{1}{k^2 + l^2} x J_1(x) \Big|_0^{R\sqrt{k^2 + l^2}} \Rightarrow \\ h(k, l) &= \frac{R}{\sqrt{k^2 + l^2}} J_1(R\sqrt{k^2 + l^2}) \end{aligned} \quad (5-11)$$

159

这个函数的定义域是整个空间的, 在每一对整数坐标系 (k, l) 上有定义。因此它对应一个无限维的序列。同时意味着这个滤波函数不能通过对图像进行线性卷积滤波实现。

• 例5.1(B) 理想带通滤波器的脉冲响应是怎样的?

理想的带通滤波器定义如下:

$$H(\mu, \nu) = \begin{cases} 1 & R_1 \leq \sqrt{\mu^2 + \nu^2} \leq R_2 \\ 0 & \text{其他} \end{cases}$$

与方框B5.1中定义的理想低通滤波器相比, 惟一的区别是式(5-9)的积分限不同:

$$\begin{aligned} h(k, l) &= \frac{1}{k^2 + l^2} \int_{R_1 \sqrt{k^2 + l^2}}^{R_2 \sqrt{k^2 + l^2}} x J_0(x) dx \\ &= \frac{1}{k^2 + l^2} x J_1(x) \Big|_{R_1 \sqrt{k^2 + l^2}}^{R_2 \sqrt{k^2 + l^2}} \\ &= \frac{1}{\sqrt{k^2 + l^2}} (R_2 J_1(R_2 \sqrt{k^2 + l^2}) - R_1 J_1(R_1 \sqrt{k^2 + l^2})) \end{aligned}$$

这是一个定义在所有 (k, l) 上的函数。因此理想带通滤波器是一个无限的脉冲响应滤波器。

例5.2(B) 理想高通滤波器的脉冲响应是怎样的?

理想的高通滤波器定义如下:

$$H(\mu, \nu) = \begin{cases} 0 & \sqrt{\mu^2 + \nu^2} \leq R \\ 1 & \text{其他} \end{cases}$$

与方框B5.1中定义的理想低通滤波器相比, 惟一的区别是式(5-9)的积分限不同:

$$\begin{aligned} h(k, l) &= \frac{1}{k^2 + l^2} \int_{R \sqrt{k^2 + l^2}}^{\infty} x J_0(x) dx \\ &= \frac{1}{k^2 + l^2} x J_1(x) \Big|_{R \sqrt{k^2 + l^2}}^{\infty} \end{aligned} \quad (5-12) \quad \boxed{160}$$

当 $x \rightarrow \infty$ 时, Bessel函数 $J_1(x)$ 趋于0, 它相当于 $\lim_{x \rightarrow \infty} J_1(x) = \frac{1}{\sqrt{x}}$ 。这意味着 x 趋于0比 $J_1(x)$ 趋

于0的速度还要快, 即 $\lim_{x \rightarrow \infty} x J_1(x) \rightarrow \infty$ 。因此, 实数域上不存在通过傅里叶变换能成为理想高通滤波器的函数。实际上, 我们可能感兴趣的最高频率是 $1/N$, 其中 N 是图像的像素个数。所以式(5-12)的上极限 ∞ 并不会出现, 这时理想的高通滤波器就变得与理想的带通滤波器一样了。

•
•
•

5.6 一维理想低通滤波器和二维理想低通滤波器之间的关系

一维理想低通滤波器是:

$$h(k) = \frac{\sin k}{k} \quad (5-13)$$

二维理想低通滤波器是:

$$h(k, l) = \frac{J_1(\sqrt{k^2 + l^2})}{\sqrt{k^2 + l^2}}$$

其中 $J_1(x)$ 是第一类1阶Bessel函数。图5-2是 $h(k)$ 关于 k 的图形, 以及当 $l = 0$ 时 $h(k, l)$ 关于 k 的图形。可以看到虽然两者看起来很相似, 但是它们有明显的细节差别: 它们有不同的零点, 而且它们波峰的振幅不相同。

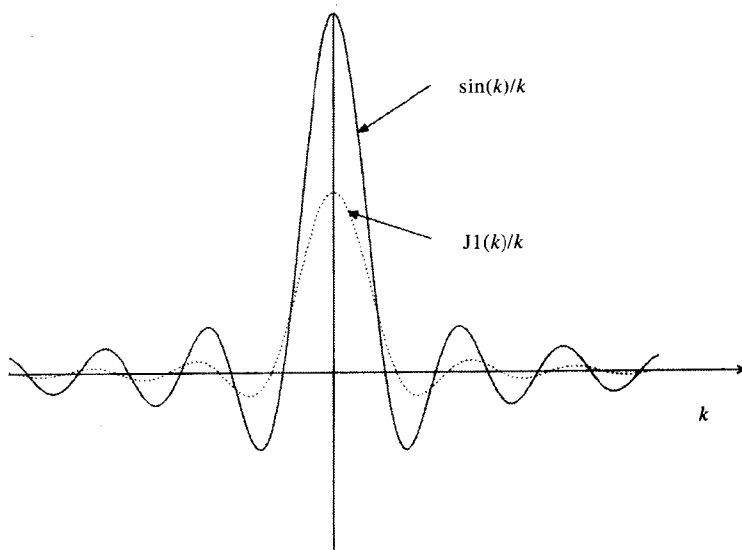


图5-2 二维理想低通滤波器的横截面与一维理想低通滤波器的横截面相似, 但却是有区别的

这意味我们不能通过把一维理想或最优(在某些标准下)的滤波器中的变量用极半径代替(即在式(5-13)中用 $\sqrt{k^2 + l^2}$ 代替 k)来构造相应的二维“理想或最优”滤波器。但是, 虽然按上述方式构造的二维滤波器不是理想的或在二维的某些标准下不是最优的, 但这种次优的滤波器却与一些最优滤波器的功效差不多。

5.7 如何实现无限域上的滤波器

实数空间的无限域上的滤波器可以通过递归来实现, 这就是所谓的递归滤波器(recursive filter)。相对应地, 实数空间的有限域上的滤波器称为非递归(non-recursive)滤波器。滤波器的表示和使用通常会用到 z 变换。

5.8 如何定义数字化一维滤波器的 z 变换

有限域上的滤波器实质上是一列有限的数字串 $\{x_1, x_2, x_3, \dots, x_n\}$ 。有时会用箭头标明串中对应于零点位置的元素。这样一串数字的 z 变换定义如下:

$$X(z) = \sum_{k=l}^m x_k z^{-k} \quad (5-14)$$

其中 l 和 m 的值是由数字串 $\{x_k\}$ 对应的 z^{-k} 中 k 为0的项确定。(例如, 如果 x^3 是 $k=0$ 对应的项, 那么在上述序列中会有 $l=-2$ 和 $m=n-3$)。

如果滤波器定义在无限域上, 则表示该滤波器的数字序列也是无限的, 它的 z 变换可以表达成无限和的形式:

$$X(z) = \sum_{k=-\infty}^{\infty} x_k z^{-k} \quad \text{或} \quad \sum_{k=0}^{\infty} x_k z^{-k} \quad \text{或} \quad \sum_{k=-\infty}^0 x_k z^{-k}$$

这时, 通常将 z 变换写成两个 z 的多项式的比值而不是像有限域情况下写成一个 z 的多项式。

161
162

5.9 为什么要用 z 变换

原因是一个以 z 变换形式定义的数字滤波器比较容易用硬件实现。在其收敛的区域, 一个序列的 z 变换惟一地定义了一个序列。更进一步的是, 它满足卷积定理: 两个序列的卷积的 z 变换等于这两个序列的 z 变换的乘积。

5.10 二维的 z 变换如何定义

设有一个二维的 $M \times N$ 的阵列, 它的 z 变换是一个关于复变量 z_1 、 z_2 的有限多项式:

$$H(z_1, z_2) = \sum_{i=0}^M \sum_{j=0}^N c_{ij} z_1^i z_2^j \quad (5-15)$$

其中 c_{ij} 是阵列的元素。对于无限阵列, 两个和变成了无限和, 而且可以写成两个有限和的比值:

$$H(z_1, z_2) = \frac{\sum_{i=0}^{M_a} \sum_{j=0}^{N_a} a_{ij} z_1^i z_2^j}{\sum_{i=0}^{M_b} \sum_{j=0}^{N_b} b_{ij} z_1^i z_2^j} \quad (5-16)$$

其中 M_a 、 N_a 、 M_b 和 N_b 是某些整数。一般来说, 取 $b_{00} = 1$ 。

B5.2 为什么一个滤波器的域决定它是否是递归的?

当将一幅图像与一个数字滤波器做卷积运算时, 实质上等价于将图像的 z 变换与滤波器的 z 变换相乘:

$$\underbrace{R(z_1, z_2)}_{\text{输出图像的}z\text{变换}} = \underbrace{H(z_1, z_2)}_{\text{滤波器的}z\text{变换}} \underbrace{D(z_1, z_2)}_{\text{输入图像的}z\text{变换}} \quad (5-17)$$

假如将式(5-16)代入式(5-17)并且将分母移到等式的左边, 则有:

$$R(z_1, z_2) \sum_{i=0}^{M_b} \sum_{j=0}^{N_b} b_{ij} z_1^i z_2^j = \left(\sum_{i=0}^{M_a} \sum_{j=0}^{N_a} a_{ij} z_1^i z_2^j \right) D(z_1, z_2)$$

163

通常 $b_{00} = 1$ 下, 在等式的左边将求和分离成 $i = 0, j = 0$ 和其他的情况:

$$R(z_1, z_2) + \left(\underbrace{\sum_{i=0}^{M_b} \sum_{j=0}^{N_b} b_{ij} z_1^i z_2^j}_{\text{并不都为0}} \right) R(z_1, z_2) = \left(\sum_{i=0}^{M_a} \sum_{j=0}^{N_a} a_{ij} z_1^i z_2^j \right) D(z_1, z_2)$$

因此:

$$R(z_1, z_2) = \left(\sum_{i=0}^{M_a} \sum_{j=0}^{N_a} a_{ij} z_1^i z_2^j \right) D(z_1, z_2) - \left(\underbrace{\sum_{i=0}^{M_b} \sum_{j=0}^{N_b} b_{ij} z_1^i z_2^j}_{\text{并不都为0}} \right) R(z_1, z_2) \quad (5-18)$$

注意 $R(z_1, z_2)$ 是系数为 r_{mn} 的 $z_1^m z_2^n$ 之和。由于 $R(z_1, z_2)$ 同时又出现在等式的右边, 因此上述公式中 r_{mn} 的值可以由先前计算的 r_{mn} 计算得到。这就是为什么要称该滤波器为递归的原因。当滤波器是有限域上的时候, b_{ij} 的值为0 (除了 $b_{00} = 1$), 因此 $R(z_1, z_2)$ 的系数 r_{mn} 此时由 a_{ij} 决定, 并且这些系数只出现在 $D(z_1, z_2)$ 中 (也就是没有递归)。

- 例5.3(B) 有一幅 256×256 的图像, 用一个无限脉冲响应滤波器对其滤波。该滤波器的 z 变换
- 可以写成变量 z_1 和 z_2 的3阶多项式与另外一个具有同样变量的3阶多项式的比值。根据输入的图像的值和滤波器的系数计算输出图像的值。

在式(5-18)中, 令 $M_a = N_a = M_b = N_b = 3$ 。那么输入图像的 z 变换是:

$$D(z_1, z_2) = \sum_{k=0}^{255} \sum_{l=0}^{255} d_{kl} z_1^k z_2^l \quad (5-19)$$

输出图像的 z 变换是

$$R(z_1, z_2) = \sum_{k=0}^{255} \sum_{l=0}^{255} r_{kl} z_1^k z_2^l \quad (5-20)$$

我们希望根据 a_{ij} 、 b_{ij} 和 d_{ij} 导出 r_{ij} 的表达式。如果将式(5-19)和式(5-20)代入式(5-18), 可以得到:

$$\sum_{k=0}^{255} \sum_{l=0}^{255} r_{kl} z_1^k z_2^l = \sum_{i=0}^3 \sum_{j=0}^3 \sum_{k=0}^{255} \sum_{l=0}^{255} (a_{ij} d_{kl} - b_{ij} r_{kl}) z_1^{k+i} z_2^{l+j} \quad (5-21)$$

为了避免单独处理第0项, 在上式中允许 $i = j = 0$ 的项存在, 但必须注意的是这时 $b_{00} = 0$ 。为了联系等式左右两端同阶项, 我们定义新的变量来对右边的等式求和。定义为:

$$\left. \begin{array}{l} k+i \equiv u \\ k-i \equiv v \end{array} \right| \Rightarrow \left. \begin{array}{l} k = \frac{u+v}{2} \\ i = \frac{u-v}{2} \end{array} \right| \quad \left. \begin{array}{l} l+j \equiv t \\ l-j \equiv s \end{array} \right| \Rightarrow \left. \begin{array}{l} l = \frac{t+s}{2} \\ j = \frac{t-s}{2} \end{array} \right|$$

接着我们必须找到新变量的上下限。下图显示了变量 k, i 的求和区域如何转换为新变量的求和区域的。变换的区域是通过变换公式确定的，可以用一系列的直线围成，即线 $i=0$ 、 $i=3$ 、 $k=0$ 和 $k=255$ ，经变换后分别得到： $v=u$ 、 $v=u-6$ 、 $v=-u$ 和 $v=510-u$ 。

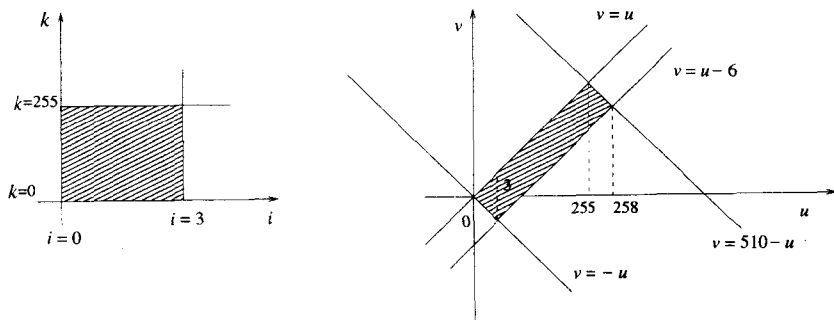


图5-3 求和区域的转换

式(5-21)变成:

$$\sum_{k=0}^{255} \sum_{l=0}^{255} r_{kl} z_1^k z_2^l = \sum_{j=0}^3 \sum_{l=0}^{255} \left\{ \sum_{u=0}^2 \sum_{v=-u}^u + \sum_{u=3}^{255} \sum_{v=u-6}^u + \sum_{u=256}^{258} \sum_{v=u-6}^{510-u} \right\} (a_{\frac{u-v}{2}} d_{\frac{u+v}{2}} - b_{\frac{u-v}{2}} r_{\frac{u+v}{2}}) z_1^u z_2^{l+j} \quad (5-22) \quad \boxed{165}$$

类似地，我们可以求得和式中变量 s 和 t 的上下限，得：

$$\begin{aligned} & \sum_{k=0}^{255} \sum_{l=0}^{255} r_{kl} z_1^k z_2^l \\ &= \left\{ \sum_{t=0}^2 \sum_{s=-t}^t \sum_{u=0}^2 \sum_{v=-u}^u + \sum_{t=0}^2 \sum_{s=-t}^t \sum_{u=3}^{255} \sum_{v=u-6}^u + \sum_{t=0}^2 \sum_{s=-t}^t \sum_{u=256}^{258} \sum_{v=u-6}^{510-u} \right. \\ &+ \sum_{t=3}^{255} \sum_{s=t-6}^t \sum_{u=0}^2 \sum_{v=-u}^u + \sum_{t=3}^{255} \sum_{s=t-6}^t \sum_{u=3}^{255} \sum_{v=u-6}^u + \sum_{t=3}^{255} \sum_{s=t-6}^t \sum_{u=256}^{258} \sum_{v=u-6}^{510-u} \\ &+ \sum_{t=256}^{258} \sum_{s=t-6}^t \sum_{u=0}^2 \sum_{v=-u}^u + \sum_{t=256}^{258} \sum_{s=t-6}^t \sum_{u=3}^{255} \sum_{v=u-6}^u + \left. \sum_{t=256}^{258} \sum_{s=t-6}^t \sum_{u=256}^{258} \sum_{v=u-6}^{510-u} \right\} \\ & \left((a_{\frac{u-v}{2}} d_{\frac{u+v}{2}} - b_{\frac{u-v}{2}} r_{\frac{u+v}{2}}) z_1^u z_2^t \right) \end{aligned} \quad (5-23)$$

进一步定义以下各式：

$$\begin{aligned} A_{ut} &\equiv \sum_{s=-t}^t \sum_{v=-u}^u \left(a_{\frac{u-v}{2}} d_{\frac{u+v}{2}} - b_{\frac{u-v}{2}} r_{\frac{u+v}{2}} \right) \\ B_{ut} &\equiv \sum_{s=-t}^t \sum_{v=u-6}^u \left(a_{\frac{u-v}{2}} d_{\frac{u+v}{2}} - b_{\frac{u-v}{2}} r_{\frac{u+v}{2}} \right) \end{aligned}$$

$$\begin{aligned}
C_{ut} &\equiv \sum_{s=-t}^t \sum_{v=u-6}^{510-u} \left(a_{\frac{u-v}{2} \frac{t-s}{2}} d_{\frac{u+v}{2} \frac{t+s}{2}} - b_{\frac{u-v}{2} \frac{t-s}{2}} r_{\frac{u+v}{2} \frac{t-s}{2}} \right) \\
D_{ut} &\equiv \sum_{s=t-6}^t \sum_{v=-u}^u \left(a_{\frac{u-v}{2} \frac{t-s}{2}} d_{\frac{u+v}{2} \frac{t+s}{2}} - b_{\frac{u-v}{2} \frac{t-s}{2}} r_{\frac{u+v}{2} \frac{t-s}{2}} \right) \\
E_{ut} &\equiv \sum_{s=t-6}^t \sum_{v=u-6}^u \left(a_{\frac{u-v}{2} \frac{t-s}{2}} d_{\frac{u+v}{2} \frac{t+s}{2}} - b_{\frac{u-v}{2} \frac{t-s}{2}} r_{\frac{u+v}{2} \frac{t-s}{2}} \right) \\
F_{ut} &\equiv \sum_{s=t-6}^t \sum_{v=u-6}^{510-u} \left(a_{\frac{u-v}{2} \frac{t-s}{2}} d_{\frac{u+v}{2} \frac{t+s}{2}} - b_{\frac{u-v}{2} \frac{t-s}{2}} r_{\frac{u+v}{2} \frac{t-s}{2}} \right) \\
G_{ut} &\equiv \sum_{s=t-6}^{510-t} \sum_{v=-u}^u \left(a_{\frac{u-v}{2} \frac{t-s}{2}} d_{\frac{u+v}{2} \frac{t+s}{2}} - b_{\frac{u-v}{2} \frac{t-s}{2}} r_{\frac{u+v}{2} \frac{t-s}{2}} \right) \\
H_{ut} &\equiv \sum_{s=t-6}^{510-t} \sum_{v=u-6}^u \left(a_{\frac{u-v}{2} \frac{t-s}{2}} d_{\frac{u+v}{2} \frac{t+s}{2}} - b_{\frac{u-v}{2} \frac{t-s}{2}} r_{\frac{u+v}{2} \frac{t-s}{2}} \right) \\
I_{ut} &\equiv \sum_{s=t-6}^{510-t} \sum_{v=u-6}^{510-u} \left(a_{\frac{u-v}{2} \frac{t-s}{2}} d_{\frac{u+v}{2} \frac{t+s}{2}} - b_{\frac{u-v}{2} \frac{t-s}{2}} r_{\frac{u+v}{2} \frac{t-s}{2}} \right)
\end{aligned} \tag{5-24}$$

则式(5-23)可改写为:

$$\begin{aligned}
\sum_{u=0}^{255} \sum_{t=0}^{255} r_{ut} z_1^u z_2^t &= \sum_{t=0}^2 \sum_{u=0}^2 A_{ut} z_1^u z_2^t + \sum_{t=0}^2 \sum_{u=3}^{255} B_{ut} z_1^u z_2^t + \sum_{t=0}^2 \sum_{u=256}^{258} C_{ut} z_1^u z_2^t \\
&+ \sum_{t=3}^{255} \sum_{u=0}^2 D_{ut} z_1^u z_2^t + \sum_{t=3}^{255} \sum_{u=3}^{255} E_{ut} z_1^u z_2^t + \sum_{t=3}^{255} \sum_{u=256}^{258} F_{ut} z_1^u z_2^t \\
&+ \sum_{t=256}^{258} \sum_{u=0}^2 G_{ut} z_1^u z_2^t + \sum_{t=256}^{258} \sum_{u=3}^{255} H_{ut} z_1^u z_2^t + \sum_{t=256}^{258} \sum_{u=256}^{258} I_{ut} z_1^u z_2^t
\end{aligned} \tag{5-25}$$

这里, 将等式左边求和变量改写为 u 、 t 。我们必须通过比较式(5-25)两边的同类项的系数得到 r_{ut} 关于 a_{ut} 、 b_{ut} 和 d_{ut} 的表达式, 但必须注意 b_{00} 在式(5-21)中定义为0。为了处理所有的情况, 我们画了图5-4中的图, 其中两个坐标轴对应式(5-25)右边的求和变量。图中的每个矩形框对应两个变量的各自取值的联合范围。

第1种情况: 当 $0 \leq u < 2$ 且 $0 \leq t < 2$ 时

$$r_{ut} = A_{ut} = \sum_{s=-t}^t \sum_{v=-u}^u \left(a_{\frac{u-v}{2} \frac{t-s}{2}} d_{\frac{u+v}{2} \frac{t+s}{2}} - b_{\frac{u-v}{2} \frac{t-s}{2}} r_{\frac{u+v}{2} \frac{t-s}{2}} \right) \tag{5-26}$$

此时定义新的求和变量:

$$i = \frac{u-v}{2}, \text{ 有 } i = u \text{ (当 } v = -u \text{)} \text{ 并且 } i = 0 \text{ (当 } v = u \text{)}。$$

$$\text{那么 } v = u - 2i \Rightarrow \frac{u+v}{2} = \frac{2u-2i}{2} = u - i$$

$j = \frac{t-s}{2}$, 有 $j = t$ (当 $s = -t$) 并且 $j = 0$ (当 $s = t$)。

$$\text{那么 } s = t - 2j \Rightarrow \frac{t+s}{2} = \frac{2t-2j}{2} = t-j$$

式5-26可改写为:

$$r_{ut} = \sum_{i=0}^u \sum_{j=0}^t (a_{ij} d_{u-i, t-j} - b_{ij} r_{u-i, t-j}) \quad (5-27)$$

我们先考虑 $u = t = 0$ 的情况, 同时注意 $b_{00} = 0$, 则:

$$r_{00} = a_{00} d_{00}$$

然后考虑 $u = 0, t = 1$, 下标 j 仅取0和1, 则:

$$r_{01} = a_{00} d_{01} + a_{01} d_{00} - b_{01} r_{00}$$

当 $u = 0, t = 2$, 下标 j 仅取0、1和2, 则:

$$r_{02} = a_{00} d_{02} + a_{01} d_{01} - b_{01} r_{01} + a_{02} d_{00} - b_{02} r_{00}$$

值得一提的是, 尽管等式右边有些 r 值出现, 但是它们的下标总是比左边 r 的下标小, 因此这些值是已经被计算出来的。

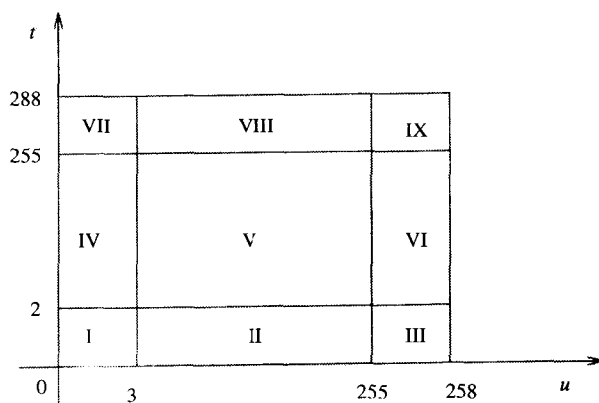


图5-4 将不同的递归公式应用于下标值的不同范围

第2种情况: $2 < u \leq 255$ 且 $0 \leq t \leq 2$

$$r_{ut} = B_{ut} = \sum_{s=-t}^t \sum_{v=u-6}^u \left(a_{\frac{u-v}{2} \frac{t-s}{2}} d_{\frac{u+v}{2} \frac{t+s}{2}} - b_{\frac{u-v}{2} \frac{t-s}{2}} r_{\frac{u+v}{2} \frac{t-s}{2}} \right) \quad (5-28)$$

与前一种情况惟一不同的是, 新的求和变量 i 将被限制为从 $i = 3$ (对应 $v = u-6$) 到 $i = 0$ (对应 $v = u$):

167
168

$$r_{ut} = \sum_{i=0}^3 \sum_{j=0}^t (a_{ij} d_{u-i, t-j} - b_{ij} r_{u-i, t-j}) \quad (5-29)$$

第3种情况: $255 < u < 258$ 且 $0 < t < 2$

在这种情况下, 因为等式左边 u 的最大值是 255, 所以等式左边没有对应的项。因此有:

$$0 = C_{ut} \equiv \sum_{s=-t}^t \sum_{v=u-6}^{510-u} \left(a_{\frac{u-v}{2} \frac{t-s}{2}} d_{\frac{u+v}{2} \frac{t+s}{2}} - b_{\frac{u-v}{2} \frac{t-s}{2}} r_{\frac{u+v}{2} \frac{t+s}{2}} \right) \quad (5-30)$$

或者

$$\sum_{i=u-255}^3 \sum_{j=0}^t (a_{ij} d_{u-i, t-j} - b_{ij} r_{u-i, t-j}) = 0 \quad (5-31)$$

考虑当 $u = 256, t = 0$ 时, 则:

$$\begin{aligned} \sum_{i=1}^3 a_{i0} d_{256-i, 0} - b_{i0} r_{256-i, 0} &= 0 \Rightarrow \\ a_{10} d_{255, 0} - b_{10} r_{255, 0} + a_{20} d_{254, 0} - b_{20} r_{254, 0} + a_{30} d_{253, 0} - b_{30} r_{253, 0} &= 0 \end{aligned} \quad (5-32)$$

$r_{255, 0}$ 、 $r_{254, 0}$ 和 $r_{253, 0}$ 的值已经通过式(5-29)计算出来。因此式(5-32)可以用于验证所计算出的值是否正确, 这就是式(5-31)的作用。而且, 我们还可以导出第4、5、6、7、8种情况。

第4种情况: $0 < u < 2$ 且 $2 < t < 255$

$$r_{ut} = D_{ut} = \sum_{s=t-6}^t \sum_{v=-u}^u \left(a_{\frac{u-v}{2} \frac{t-s}{2}} d_{\frac{u+v}{2} \frac{t+s}{2}} - b_{\frac{u-v}{2} \frac{t-s}{2}} r_{\frac{u+v}{2} \frac{t+s}{2}} \right) \quad (5-33)$$

或者

$$r_{ut} = \sum_{i=0}^u \sum_{j=0}^3 (a_{ij} d_{u-i, t-j} - b_{ij} r_{u-i, t-j}) \quad (5-34)$$

第5种情况: $2 < u < 255$ 且 $2 < t < 255$

$$r_{ut} = E_{ut} = \sum_{s=t-6}^t \sum_{v=u-6}^u \left(a_{\frac{u-v}{2} \frac{t-s}{2}} d_{\frac{u+v}{2} \frac{t+s}{2}} - b_{\frac{u-v}{2} \frac{t-s}{2}} r_{\frac{u+v}{2} \frac{t+s}{2}} \right) \quad (5-35)$$

或者

$$r_{ut} = \sum_{i=0}^3 \sum_{j=0}^3 (a_{ij} d_{u-i, t-j} - b_{ij} r_{u-i, t-j}) \quad (5-36)$$

第6种情况: $255 < u \leq 258$ 且 $2 < t \leq 255$

$$0 = F_{ut} = \sum_{s=t-6}^t \sum_{v=u-6}^{510-u} \left(a_{\frac{u-v}{2} \frac{t-s}{2}} d_{\frac{u+v}{2} \frac{t+s}{2}} - b_{\frac{u-v}{2} \frac{t-s}{2}} r_{\frac{u+v}{2} \frac{t+s}{2}} \right) \quad (5-37) \quad \boxed{169}$$

或者

$$\sum_{i=3}^{u-255} \sum_{j=0}^3 (a_{ij} d_{u-i, t-j} - b_{ij} r_{u-i, t-j}) = 0 \quad (5-38)$$

第7种情况: $0 < u \leq 2$ 且 $255 < t \leq 258$

$$0 = G_{ut} = \sum_{s=t-6}^{510-t} \sum_{v=-u}^u \left(a_{\frac{u-v}{2} \frac{t-s}{2}} d_{\frac{u+v}{2} \frac{t+s}{2}} - b_{\frac{u-v}{2} \frac{t-s}{2}} r_{\frac{u+v}{2} \frac{t+s}{2}} \right) \quad (5-39)$$

或者

$$\sum_{i=0}^u \sum_{j=3}^{t-255} (a_{ij} d_{u-i, t-j} - b_{ij} r_{u-i, t-j}) = 0 \quad (5-40)$$

第8种情况: $2 < u \leq 255$ 且 $255 < t \leq 258$

$$0 = H_{ut} = \sum_{s=t-6}^{510-t} \sum_{v=u-6}^u \left(a_{\frac{u-v}{2} \frac{t-s}{2}} d_{\frac{u+v}{2} \frac{t+s}{2}} - b_{\frac{u-v}{2} \frac{t-s}{2}} r_{\frac{u+v}{2} \frac{t+s}{2}} \right) \quad (5-41)$$

或者

$$\sum_{i=0}^3 \sum_{j=3}^{t-255} (a_{ij} d_{u-i, t-j} - b_{ij} r_{u-i, t-j}) = 0 \quad (5-42)$$

第9种情况: $2 < u \leq 255$ 并且 $255 < t \leq 258$

$$0 = I_{ut} = \sum_{s=t-6}^{510-t} \sum_{v=u-6}^{510-u} \left(a_{\frac{u-v}{2} \frac{t-s}{2}} d_{\frac{u+v}{2} \frac{t+s}{2}} - b_{\frac{u-v}{2} \frac{t-s}{2}} r_{\frac{u+v}{2} \frac{t+s}{2}} \right) \quad (5-43)$$

或者

$$\sum_{i=0}^3 \sum_{j=3}^{t-255} (a_{ij} d_{u-i, t-j} - b_{ij} r_{u-i, t-j}) = 0 \quad (5-44) \quad \vdots$$

5.11 一维递归滤波器和二维递归滤波器有什么基本的区别

在一维情况下, 如果把 z 变换写成两个多项式的比值时, 可以通过对这些多项式因式分解来获得该变换的极点和零点。只要知道极点, 就可以计算出任何在 z 变换收敛区域上的 z 变

换的逆。

在二维情况下,多项式的因式分解困难多了,即使我们可以这样做,分母多项式的零点(极点)通常不是孤立的离散点,而是变量的连续值的区域。而且, (z_1, z_2) 是一个四维空间,这就不可能确定多项式的收敛区域了。

170

5.12 如何知道滤波器没有扩大噪声

一个滤波器必须是稳定(stable)的,它不能放大噪声。称一个数字滤波器是稳定的,如果有界输入的输出也是有界的。一个数字滤波器 $h(n, m)$ 是稳定的当且仅当:

$$\sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} |h(n, m)| < \infty$$

如果滤波器是有限域上的或不是递归的,那么上面的准则显然是可以实现的。对于无限脉冲响应(递归)滤波器,我们必须利用有关的定理,仔细分析滤波器的性能。

5.13 是否有另一种方法使用无限脉冲响应滤波器

是的,可以减弱对滤波器频率响应的要求。特别是,可以采用以下方式处理:

1) 确定滤波器的系统函数(频率响应)。

2) 选择一个有限滤波器(非递归的),使它尽可能逼近所要求的系统函数。

这种方法导致逼近理论(approximation theory)的产生。

5.14 为什么需要逼近理论

因为所要求的滤波器频率响应不可能用一个有限脉冲响应滤波器来实现。

5.15 如何知道一个近似滤波器是否性能良好

我们使用目标响应与近似响应之差的Chebyshev范数(Chebyshev norm)去衡量逼近的质量,它可以被看做是这两个函数的最大偏离。

假设 $F(\mu, \nu)$ 是所要求的理想频率响应, $H(\mu, \nu)$ 是有限近似滤波器的频率响应。那么 $F(\mu, \nu)$ 与 $H(\mu, \nu)$ 的近似程度可以通过以下衡量:

$$\text{误差} = \|F(\mu, \nu) - H(\mu, \nu)\| \equiv \max_{(\mu, \nu)} |F(\mu, \nu) - H(\mu, \nu)| \quad (5-45)$$

5.16 对一个给定的理想系统函数的最佳逼近是什么

理想系统函数 $F(\mu, \nu)$ 的最佳逼近就是使得所求 $H(\mu, \nu)$ 与理想系统函数误差最小化,即

171

$$\text{最佳逼近误差} = \min_{H(\mu, \nu)} \{ \max_{(\mu, \nu)} |F(\mu, \nu) - H(\mu, \nu)| \} \quad (5-46)$$

5.17 为什么根据Chebyshev范数而不是均方误差来评价一个近似值

总体均方误差 $(\iint [F(\mu, \nu) - H(\mu, \nu)]^2 d\mu d\nu)$ 是 $F(\mu, \nu)$ 与 $H(\mu, \nu)$ 接近程度的平均值。两个不同

的逼近可能有同样的均方误差,但是可以例举出其中一个在特定频率上与滤波函数有极不相符特征(如图5-5所示)。而Chebyshev范数能保证不会出现使两个函数的偏离超过特定限制的频率值。

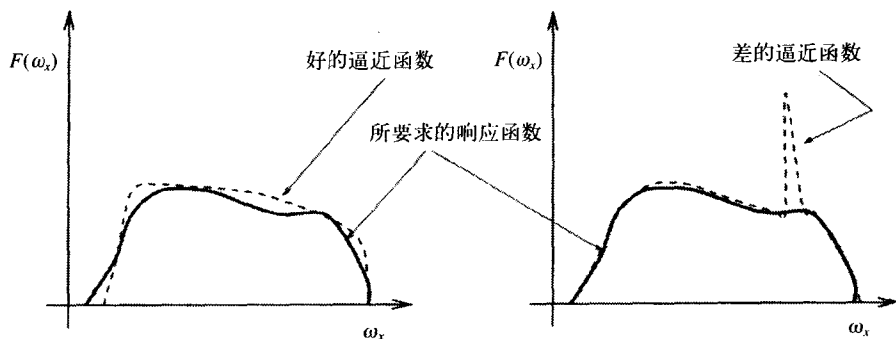


图5-5 在频域上,一个好的逼近是它不会偏离所要求的响应函数太多

5.18 如何获得一个系统函数的逼近

设计一个二维滤波器有3种方法。所有3种方法是直接对一维滤波器中相应方法的改动。它们是:

- 窗口法
- 频率采样
- 线性规划

以上方法中,只有最后一种方法可以给出最佳的逼近,它可以使得逼近函数和精确函数的最大偏差最小化。

5.19 什么是窗口法

使得一个无限脉冲响应滤波器有限化最直接的方法是将其截断成所需的长度。直观上,这种方法是把一个窗体放在滤波器上,而只观察窗口里面的信息,因此这种方法就叫窗口法(windowing)。然而,这种方法会使得滤波器的边缘出现锐化。

172

5.20 窗口法存在什么问题

对于一个不连续的函数(类似一个尖锐的截尾滤波器),它的傅里叶变换的高频部分会有许多的波幅,这是因为那些尖锐边缘的变换会产生许多高频谐波分量。因此,截断会造成系统滤波函数高频信息的失真。

5.21 如何提高窗口法处理的效果

对一个滤波函数在特定频域范围内的截断相当于使它与一个窗口函数相乘,窗口函数的形式为:

$$w(x, y) = \begin{cases} 1 & \text{当 } -a \leq x \leq a \text{ 且 } -b \leq y \leq b \\ 0 & \text{其他情况} \end{cases}$$

其中 a, b 为某些参数。为了避免由截断滤波器产生的边界的锐化, 这个窗口函数必须用一个光滑函数代替。对于一维的信号, 已经有几种光滑的窗口了。

5.22 如何用一维信号的窗口函数来定义图像的窗口函数

一旦我们能够在二维情况下找到一个好的窗口函数, 将它延拓到二维的情形并非易事。必须清楚的是即使 $w(x)$ 是一个好的一维窗口函数, 它的圆对称函数 $w(r) \equiv w(\sqrt{x^2 + y^2})$ 的频率响应与 $w(x)$ 并不是一致的。换句话说, 圆对称的窗口函数的傅里叶变换并不是一维窗口函数傅里叶变换的圆对称函数。在本章的前几节已经有很多类似的讨论了。(参见图5-2)。

然而, 一般而言, 如果 $w(x)$ 是一个好的一维窗口函数, 那么 $w_2(\sqrt{x^2 + y^2})$ 也是一个好的二维圆对称窗口函数。在这两种情况下都必须注意, 当对窗口函数离散化时, 必须取足够多的采样点以避免混淆现象的发生。

窗口方法或许可以得到一些相当好的二维滤波器, 但是这些滤波器都不是所需要的滤波器的最优逼近, 因为我们并没有做任何最优化逼近的工作。

5.23 所要解决的逼近问题的形式定义是什么

假设 $F(\mu, \nu)$ 是一个理想的、不可实现的滤波响应。目标是要找到一组系数 c_1, c_2, \dots, c_n 来生成一个多项式 $P(\mu, \nu) \equiv \sum_{i=1}^n c_i g_i(\mu, \nu)$, 使得误差函数 Δ Chebyshev范数达到最小, 其中 Δ 定义为:

$$\Delta \equiv F(\mu, \nu) - \sum_{i=1}^n c_i g_i(\mu, \nu) \quad (5-47)$$

这里 $g_i(\mu, \nu)$ 是已知的关于频率 μ 和 ν 的函数, 比如 $\cos(\mu + \nu)$ 、 $\sin(2\mu + \nu)$ 等等。

在式(5-47)中, μ 和 ν 取连续值。然而, 我们需考虑有限采样集下的误差, 即离散情况下在 (μ, ν) 空间取样 M 个点的误差函数 Δ :

$$\Delta_m \equiv F(\mu_m, \nu_m) - \sum_{i=1}^n c_i g_i(\mu_m, \nu_m) \quad m = 1, 2, \dots, M \quad (5-48)$$

我们的目标是选择系数 c_1, c_2, \dots, c_n 使得 Δ_m 的最大值尽可能的小, 这可以通过线性规划(linear programming)的方法求解这个问题。

5.24 什么是线性规划

线性规划方法是用于解决以下问题的最优化方法: 选择 n_1 个正的变量 x_{1i} 和 n_2 个变量 x_{2j} 使得

$$\text{最小化: } z = \mathbf{c}_1^T \mathbf{x}_1 + \mathbf{c}_2^T \mathbf{x}_2 + d$$

$$\text{满足: } \mathbf{A}_{11} \mathbf{x}_1 + \mathbf{A}_{12} \mathbf{x}_2 \geq \mathbf{B}_1 \quad p_1 \text{ 个不等式约束}$$

$$\mathbf{A}_{21} \mathbf{x}_1 + \mathbf{A}_{22} \mathbf{x}_2 = \mathbf{B}_2 \quad p_2 \text{ 个等式约束}$$

$$\begin{array}{ll} x_{1i} \geq 0 & n_1 \text{ 个变量 } x_1 \\ x_{2j} \text{ 是自由的} & n_2 \text{ 个变量 } x_2 \end{array}$$

其中 c_1 、 c_2 、 B_1 、 B_2 是向量, A_{11} 、 A_{12} 、 A_{21} 、 A_{22} 是矩阵。向量 x_1 、 x_2 分别由 x_{1i} 和 x_{2j} 组成。

我们可以用商业的算法包求解上面的问题。因此可以简单地使用上面的方法说明该问题, 然后用商业的算法包来求解。

5.25 如何把滤波器设计问题转化为线性规划问题

假设 $F(\mu, \nu)$ 是滤波器的目标频率响应, $h(n, m)$ 是要使用的数字化的有限滤波器。那么数字化滤波器的频率响应是:

$$H(\mu, \nu) = \sum_{n=-N}^N \sum_{m=-N}^N h(n, m) e^{-j(\mu n + \nu m)} \quad (5-49)$$

其中 $h(n, m)$ 是实数。从 B5-3 的方框中可以看到, 如果 $h(\mu, \nu) = h(-\mu, -\nu)$, 那么系统函数是实的而且可以表达成:

$$H(\mu, \nu) = 2 \sum_{n=-N}^N \sum_{m=1}^N h(n, m) \cos(\mu n + \nu m) + 2 \sum_{n=1}^N h(n, 0) \cos(\mu n) + h(0, 0) \quad (5-50) \quad [174]$$

这个等式表达了在实域上离散傅里叶变换的形式。我们想确定这些值, 也就是想确定 $h(0, 0)$, $h(1, 0)$, \dots , $h(N, 0)$, \dots , $h(-N, m)$, $h(-N+1, m)$, \dots , $h(N, m)$, $m = 1, 2, \dots, N$, 使得 $H(\mu, \nu)$ 最优地逼近目标滤波响应 $F(\mu, \nu)$ 。这意味着我们要选择上述 $h(i, j)$ 的值, 使得:

$$\delta \equiv \max_{(\mu, \nu)} |H(\mu, \nu) - F(\mu, \nu)| \quad (5-51)$$

达到最小。

变量 (μ, ν) 取连续值。我们在 (μ, ν) 中选择 p_1 个具体的点作为约束点。对于每一点 (μ_i, ν_i) , $|H(\mu_i, \nu_i) - F(\mu_i, \nu_i)|$ 的值都应小于或等于 δ 。所以, 我们可以得到每一点的条件是:

$$\delta \geq |H(\mu_i, \nu_i) - F(\mu_i, \nu_i)| \Rightarrow \begin{cases} -\delta \leq H(\mu_i, \nu_i) - F(\mu_i, \nu_i) \\ \delta \geq H(\mu_i, \nu_i) - F(\mu_i, \nu_i) \end{cases}$$

或者

$$\left. \begin{array}{l} \delta + H(\mu_i, \nu_i) \geq F(\mu_i, \nu_i) \\ \delta - H(\mu_i, \nu_i) \geq -F(\mu_i, \nu_i) \end{array} \right\} \text{ 当 } i = 1, 2, \dots, p_1 \text{ 时} \quad (5-52)$$

问题之一是: 在式(5-52)中 $2p_1$ 个不等式约束下, 尝试最小化 δ , 其中 $\delta > 0$, 同时 $h(i, j)$ 是自由的, 可以是正的, 也可以是负的。

从式(5-50)可知, 要选择的变量个数是

$$(2N+1)N + N + 1 = 2N^2 + N + N + 1 = 2N^2 + 2N + 1$$

因此, 有 $2N^2 + 2N + 1$ 个自由变量和一个约束变量 (即 δ 本身)。没有等式约束 ($p_2 = 0$), 而有 $2p_1$ 个不等式约束。线性规划表达为:

最小化: $z = x_{11}$

$$\begin{array}{lll}
 \text{满足: } A_{11}x_{11} + A_{12}x_2 \geq B_1 & 2p_1 \text{ 个不等式约束} & \\
 x_{11} \geq 0 & 1 \text{ 个非负变量} & (5-53) \\
 x_{2j} \text{ 是自由的} & 2N^2 + 2N + 1 \text{ 个自由变量} &
 \end{array}$$

175

其中

$$\begin{aligned}
 x_{11} &= \delta \\
 x_2 &= \begin{pmatrix} h(0,0) \\ h(1,0) \\ h(2,0) \\ \vdots \\ h(N,0) \\ h(-N,1) \\ \vdots \\ h(N,1) \\ h(-N,2) \\ \vdots \\ h(N,2) \\ \vdots \\ h(-N,N) \\ \vdots \\ h(N,N) \end{pmatrix} \leftarrow \begin{matrix} (2N^2 + 2N + 1) \times 1 \\ \text{矩阵} \end{matrix} \quad B_1 = \begin{pmatrix} F(\mu_1, \nu_1) \\ -F(\mu_1, \nu_1) \\ F(\mu_2, \nu_2) \\ -F(\mu_2, \nu_2) \\ \vdots \\ F(\mu_{p_1}, \nu_{p_1}) \\ -F(\mu_{p_1}, \nu_{p_1}) \end{pmatrix} \leftarrow \begin{matrix} 2p_1 \times 1 \\ \text{矩阵} \end{matrix} \\
 A_{11} &= \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \leftarrow \begin{matrix} 2p_1 \times 1 \\ \text{矩阵} \end{matrix}
 \end{aligned}$$

A_{12} 是一个维数为 $2p_1 \times (2N^2 + 2N + 1)$ 的矩阵, 其中的元素可以通过将式(5-50)代入式(5-52)或式(5-53)得到, 即可以表示成:

$$A_{12} = \begin{pmatrix} 1 & 2 \cos(\mu_1) & 2 \cos(2\mu_1) & \dots & 2 \cos(N\mu_1) & 2 \cos(-N\mu_1 + \nu_1) & \dots \\ -1 & -2 \cos(\mu_1) & -2 \cos(2\mu_1) & \dots & -2 \cos(N\mu_1) & -2 \cos(-N\mu_1 + \nu_1) & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 2 \cos(\mu_{p_1}) & 2 \cos(2\mu_{p_1}) & \dots & 2 \cos(N\mu_{p_1}) & 2 \cos(-N\mu_{p_1} + \nu_{p_1}) & \dots \\ -1 & -2 \cos(\mu_{p_1}) & -2 \cos(2\mu_{p_1}) & \dots & -2 \cos(N\mu_{p_1}) & -2 \cos(-N\mu_{p_1} + \nu_{p_1}) & \dots \\ \\ 2 \cos(N\mu_1 + \nu_1) & \dots & 2 \cos(-N\mu_1 + N\nu_1) & \dots & 2 \cos(N\mu_1 + N\nu_1) & \dots & \\ -2 \cos(N\mu_1 + \nu_1) & \dots & -2 \cos(-N\mu_1 + N\nu_1) & \dots & -2 \cos(N\mu_1 + N\nu_1) & \dots & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 2 \cos(N\mu_{p_1} + \nu_{p_1}) & \dots & 2 \cos(-N\mu_{p_1} + N\nu_{p_1}) & \dots & 2 \cos(N\mu_{p_1} + N\nu_{p_1}) & \dots & \\ -2 \cos(N\mu_{p_1} + \nu_{p_1}) & \dots & -2 \cos(-N\mu_{p_1} + N\nu_{p_1}) & \dots & -2 \cos(N\mu_{p_1} + N\nu_{p_1}) & \dots & \end{pmatrix}$$

176

当我们把问题用线性规划的方式来表达, 那么商业的线性规划软件包就可以用来求解。直接运用线性规划的一个缺陷是, 对于给定的所需系数的数目 (如给定 N), 需要大量的约束点。但是, 最后求得的滤波器是在Chebyshev下最优的。

B5.3可以证明 如果滤波器的脉冲响应是对称的, 即 $h(n, m) = h(-n, -m)$, 那么系统函数是实的, 而且可以写成式(5-50)的形式。

由式(5-49), 有:

$$H(\mu, \nu) = \sum_{n=-N}^N \sum_{m=-N}^N h(n, m) [\cos(\mu n + \nu m) + j \sin(\mu n + \nu m)] \quad (5-54)$$

只考虑虚部, 并且将对 m 的求和分成大于、等于、小于0三种情况:

$$\begin{aligned} \sum_{n=-N}^N \sum_{m=-N}^N h(n, m) \sin(\mu n + \nu m) = \\ \sum_{n=-N}^N \left[\sum_{m=-N}^{-1} h(n, m) \sin(\mu n + \nu m) + \sum_{m=1}^N h(n, m) \sin(\mu n + \nu m) + h(n, 0) \sin(\mu n) \right] \end{aligned}$$

在括弧中 m 为负的求和部分, 将求和变量变为 $\tilde{m} = -m$, 则:

$$\begin{aligned} \sum_{n=-N}^N \sum_{m=-N}^N h(m, n) \sin(\mu n + \nu m) = \sum_{n=-N}^N \left[\sum_{\tilde{m}=N}^1 h(n, -\tilde{m}) \sin(\mu n - \nu \tilde{m}) \right. \\ \left. + \sum_{m=1}^N h(n, m) \sin(\mu n + \nu m) + h(n, 0) \sin(\mu n) \right] \end{aligned}$$

我们同样可以用变量替换的方法来处理对 n 的负值部分的求和, $\tilde{n} = -n$ 。则:

$$\begin{aligned} \sum_{n=-N}^N \sum_{m=-N}^N h(m, n) \sin(\mu n + \nu m) = \underbrace{\sum_{\tilde{n}=N}^1 \sum_{\tilde{m}=1}^N h(-\tilde{n}, -\tilde{m}) \sin(-\mu \tilde{n} - \nu \tilde{m})}_{\text{虚部}} \\ + \sum_{\tilde{n}=1}^N \sum_{m=1}^N h(-\tilde{n}, m) \sin(-\mu \tilde{n} + \nu m) + \sum_{n=1}^N \sum_{\tilde{m}=1}^N h(n, -\tilde{m}) \sin(\mu n - \nu \tilde{m}) \\ + \underbrace{\sum_{n=1}^N \sum_{m=1}^N h(n, m) \sin(\mu n + \nu m)}_{\text{实部}} + \underbrace{\sum_{n=1}^N h(n, 0) \sin(\mu n)}_{\text{实部}} \\ + \underbrace{\sum_{\tilde{n}=1}^N h(-\tilde{n}, 0) \sin(-\mu \tilde{n})}_{\text{虚部}} + \underbrace{\sum_{\tilde{m}=1}^N h(0, -\tilde{m}) \sin(-\nu \tilde{m})}_{\text{虚部}} \\ + \underbrace{\sum_{m=1}^N h(0, m) \sin(\nu m)}_{\text{虚部}} + \underbrace{h(0, 0) \sin(\mu 0)}_{\text{实部}} = 0 \end{aligned} \quad (5-55)$$

在上述表达式中, 我们也对 $m = 0$ 或 $n = 0$ 做了分离处理。因为:

$$h(n, m) = h(-n, -m) \Rightarrow \begin{cases} h(0, m) = h(0, -m) \\ h(n, 0) = h(-n, 0) \\ h(n, -m) = h(-n, m) \end{cases} \quad (5-56)$$

而且因为 $\sin(-x) = -\sin(x)$, 所以式(5-55)中相同类型下划线的部分互相抵消。因此式(5-54)的虚部是0, 即滤波器傅里叶变换是实的, 可得到:

$$H(\mu, \nu) = \sum_{n=-N}^N \sum_{m=-N}^N h(n, m) \cos(\mu n + \nu m) \quad (5-57)$$

通过对负下标的部分做变量替换, 可以更进一步简化表达式 (即对于负的 n , 令 $\tilde{n} = -n$; 对于负的 m , 令 $\tilde{m} = -m$):

$$\begin{aligned} H(\mu, \nu) &= \\ &= \sum_{n=-N}^N \left[\sum_{\tilde{m}=N}^1 h(n, -\tilde{m}) \cos(\mu n - \nu \tilde{m}) \right. \\ &\quad \left. + \sum_{m=1}^N h(n, m) \cos(\mu n + \nu m) + h(n, 0) \cos(\mu n) \right] \\ &= \underbrace{\sum_{\tilde{n}=N}^1 \sum_{\tilde{m}=N}^1 h(-\tilde{n}, -\tilde{m}) \cos(-\mu \tilde{n} - \nu \tilde{m})}_{\text{下划线部分}} + \sum_{\tilde{n}=N}^1 \sum_{m=1}^N h(-\tilde{n}, m) \cos(-\mu \tilde{n} + \nu m) \\ &\quad + \underbrace{\sum_{n=1}^N \sum_{\tilde{m}=N}^1 h(n, -\tilde{m}) \cos(\mu n - \nu \tilde{m})}_{\text{下划线部分}} + \sum_{n=1}^N \sum_{m=1}^N h(n, m) \cos(\mu n + \nu m) \\ &\quad + \underbrace{\sum_{\tilde{n}=N}^1 h(-\tilde{n}, 0) \cos(-\mu \tilde{n})}_{\text{下划线部分}} + \underbrace{\sum_{n=1}^N h(n, 0) \cos(\mu n)}_{\text{下划线部分}} \\ &\quad + \underbrace{\sum_{\tilde{m}=N}^1 h(0, -\tilde{m}) \cos(-\tilde{m} \nu)}_{\text{下划线部分}} + \underbrace{\sum_{m=1}^N h(0, m) \cos(m \nu)}_{\text{下划线部分}} + h(0, 0) \end{aligned}$$

再由式(5-56)和 $\cos(-x) = \cos x$, 对上面的等式的下划线部分合并同类项, 得到:

$$\begin{aligned} H(\mu, \nu) &= 2 \sum_{n=1}^N \sum_{m=1}^N h(n, m) \cos(\mu n + \nu m) + 2 \sum_{n=1}^N \sum_{m=1}^N h(-n, m) \cos(-\mu n + \nu m) \\ &\quad + 2 \sum_{m=1}^N h(0, m) \cos(m \nu) + h(0, 0) + 2 \sum_{n=1}^N h(n, 0) \cos(\mu n) \end{aligned}$$

最后, 将前三项写在一起, 可得到:

$$H(\mu, \nu) = 2 \sum_{n=-N}^N \sum_{m=1}^N h(n, m) \cos(\mu n + \nu m) + 2 \sum_{n=1}^N h(n, 0) \cos(\mu n) + h(0, 0)$$

5.26 如何减少线性规划求解的计算强度

有一些这样的方法，在迭代算法的框架内可以把问题分割成许多小的问题。在算法的每个迭代步骤中，最优解是建立在约束点的子集上的。在每个迭代步骤算法都使用线性规划，并且缩小Chebyshev误差的范围。因此，它把一个大规模的线性规划问题分割成许多小的部分，更容易操作了。

5.27 迭代方法的主要思想是什么

假设我们有一个金属薄片，想用它覆盖一个曲面。每次我们试图弯曲薄片使它吻合表面的某些点，而在其他点由于其韧性会突起。我们发现很难使得每个点都很吻合。因此我们首先将薄片弯曲或扭曲以使它可以吻合某些点（如图5-6所示），然后我们找那些偏离目标形状最大的点，再次对薄片进行变形使得它可以吻合另一部分点，其中包括原先最大偏离的点。但显然现在的薄片还不能吻合目标形状的所有点。因为它包含了剩余点中偏离最大的点，所以新的薄片吻合目标形状的程度会比原先的差一点。换句话说，随着吻合变形处理的进行，吻合误差逐渐增加，但金属薄片与目标形状的吻合会越来越好。因为从一步迭代到下一步迭代之间，吻合误差的下限都会增加，所以称这样一个算法为极大化算法(maximizing)。

179

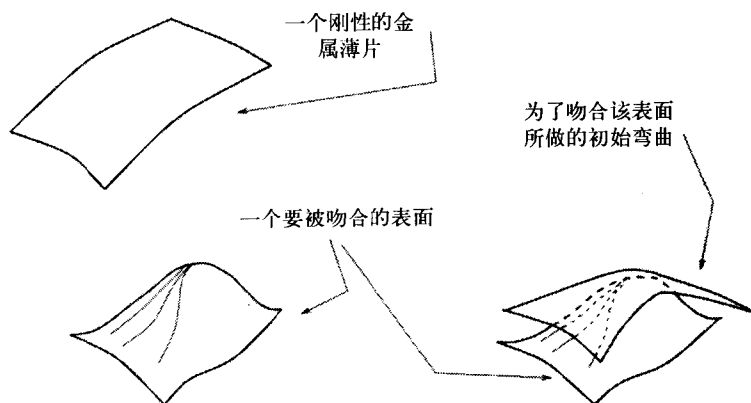


图5-6 如何跟一个表面逐渐吻合

5.28 有什么算法可以减少吻合误差的上限吗

是的，有一种叫极小化方法(minimizing)。还有一种叫极小-极大化方法(mini-max)，即同时增加误差的下限和减少误差的上限。在这里我们不打算讨论这些方法。

5.29 最大化算法是如何工作的

利用等式的极限集合(limiting set of equation)的概念和La Vallee Poussin定理就能处理了。

5.30 什么是等式的极限集合

式(5-48)的集合称为是有极限 (limiting) 的, 如果所有 Δ_m 的值不为0而且所有绝对值 $|\Delta_m|$ 不会随参数 c 的选择同步减小。

5.31 什么是La Vallee Poussin定理

180

假设式(5-48)生成一个极限集合。我们根据Chebyshev准则选择一些能最佳逼近 F 的 c_i 值。记这种逼近为 $P^* = \sum_{i=1}^n c_i^* g_i(\mu, \nu)$ 。根据Chebyshev准则, 在有 M 个点 $(\mu_1, \nu_1), (\mu_2, \nu_2), \dots, (\mu_m, \nu_m)$ 的集合 X 上最佳逼近意味着:

$$\underbrace{\max |P^* - F|}_{\text{在点X的集合上}} \leq \underbrace{\max |P - F|}_{\text{在同样的点集X上}}$$

其中 P 是任何其他逼近, 即由任何 c_i 的其他参数集确定的逼近。一般的La Vallee Poussin定理规定上述最佳Chebyshev逼近误差比随机逼近 p 的误差的最小值大。

$$\min_X |P - F| \leq \underbrace{\max_X |P^* - F|}_{\text{最佳Chebyshev逼近误差}}$$

所以, 定理表述为:

$$\min_X |P - F| \leq \max_X |P^* - F| \leq \max_X |P - F| \quad (5-58)$$

等价于最佳Chebyshev逼近误差有上下界, 分别是其他逼近误差的最小和最大值。

5.32 如何证明La Vallee Poussin定理

式(5-58)的右边显然由最佳Chebyshev逼近的定义可以得到。为了证明左边的不等式, 我们首先假设它不成立。假设:

$$\max_X |P^* - F| < \min_X |P - F| \Rightarrow \max_X |\Delta_m^*| < \min_X |\Delta_m|$$

这意味着所有 Δ_m^* 的值要比所有 Δ_m 的值小 (因为 Δ_m^* 的最大值比 Δ_m 最小值要小)。这意味着如果我们改变参数 c_i 的值为 c_i^* 的值, 那么所有的 Δ_m 的绝对值同步下降。这与我们在点集 X 上等式集合是有极限的假设相矛盾。所以式(5-58)是对的。

5.33 迭代算法的步骤是怎样的

假设我们选择一个点集 X 的子集 X_k (X 中的离散点 (μ, ν) 是被选择来确定最佳逼近的)。在这个子集上, 我们选择出的最佳逼近为:

$$P_k(\mu, \nu) = \sum_{i=1}^n c_i^k g_i(\mu, \nu) \text{ 在 } X_k \text{ 中最好}$$

而且, 定义在 X_k 上的误差为:

181

$$\delta_k \equiv \max_{X_k} |P_k - F|$$

在整个 X 上的误差定义为:

$$E_k \equiv \max_X |P_k - F|$$

现在寻求一个新的点集,它是 X 的子集,使得 P_k 在 X_{k+1} 上有最大误差

$$\max_{X_{k+1}} |P_k - F| = E_k$$

最小的误差

$$\min_{X_{k+1}} |P_k - F| \geq \delta_k$$

并且,等式的集合

$$\Delta \equiv F(\mu, \nu) - \sum_{i=1}^n c_i g_i(\mu, \nu) \quad \forall (\mu, \nu) \in X_{k+1}$$

是有极限的。根据La Vallee Poussin定理,有:

$$\min_{X_{k+1}} |P_k - F| \leq \max_{X_{k+1}} |P_{k+1} - F| \leq \max_{X_{k+1}} |P_k - F|$$

$$\delta_k \leq \delta_{k+1} \leq E_k \quad (5-59)$$

其中 P_{k+1} 是新点集 X_{k+1} 中的最佳逼近,并且:

$$\delta_{k+1} = \max_{X_{k+1}} |P_{k+1} - F|$$

从选择方式可以看到,每次额外的点都包含在子集 X_k 中,显然通过下限的增大,迭代后式(5-59)的两个不等式会缩窄。

5.34 可以逼近一个在频域上有效的滤波器吗

答案是肯定的,可以通过系统函数在其他频率值的函数来计算它在特定频率的值。换句话说, $H(\mu, \nu)$ 可以通过 $H(k, l)$ 表示,其中 k, l 取特定的整数值,并计算 $H(k, l)$ 的值使得 $\max |H(\mu, \nu) - F(\mu, \nu)|$ 最小。我们将考虑特定点 (μ, ν) 的离散值,这些点就是约束点。

5.35 如何构造一个函数去实现用滤波器的其他频率值表达它的某些频率的值

考虑用于表达滤波器的脉冲响应与系统函数关系的等式:

$$H(\mu, \nu) = \frac{1}{2N+1} \sum_{n=-N}^N \sum_{m=-N}^N h(n, m) e^{-j(\frac{\mu n + \nu m}{2N+1})2\pi} \quad (5-60) \quad \boxed{182}$$

注意到 (μ, ν) 取整数值,并且本章到此为止都没有用它们来表示有角的频率。这个系统函数的离散傅里叶逆变换是:

$$h(m, n) = \frac{1}{2N+1} \sum_{k=-N}^N \sum_{l=-N}^N H(k, l) e^{j\frac{2\pi}{2N+1}(nk+ml)} \quad (5-61)$$

令 $2N+1 \equiv p$ 。注意如果假设 $H(k, l) = H(-k, -l)$ ，这时虚部正好可以消掉，那么滤波函数是实的（这也是我们想要的），即

$$h(m, n) = \frac{1}{p} \sum_{k=-N}^N \sum_{l=-N}^N H(k, l) \cos \left[\frac{2\pi}{p} (nk + ml) \right] \quad (5-62)$$

由式(5-62)，显然有 $h(m, n) = h(-m, -n)$ ，并且利用式(5-60)可以得到：

$$H(\mu, \nu) = \frac{1}{p} \sum_{n=-N}^N \sum_{m=-N}^N h(n, m) \cos \left[\frac{2\pi}{p} (\mu n + \nu m) \right] \quad (5-63)$$

为了用 H 的其他值表达 $H(\mu, \nu)$ ，将式(5-62)代入式(5-63)：

$$H(\mu, \nu) = \frac{1}{p^2} \sum_{n=-N}^N \sum_{m=-N}^N \sum_{k=-N}^N \sum_{l=-N}^N H(k, l) \cos \left[\frac{2\pi}{p} (nk + ml) \right] \cos \left[\frac{2\pi}{p} (\mu n + \nu m) \right] \quad (5-64)$$

可以证明上式可以被改写为（参见方框B5.4）：

$$H(\mu, \nu) = \frac{1}{(2N+1)^2} \left[\frac{1}{2} H(0, 0) \Phi(0, 0) + \sum_{l=1}^N H(0, l) \Phi(0, l) + \sum_{k=1}^N \sum_{l=-N}^N H(k, l) \Phi(k, l) \right] \quad (5-65)$$

其中 $\Phi(k, l)$ 是已知的三角函数。

- 例5.4 构造一个 3×3 的，具有以下频率响应的实对称的滤波器（所谓对称就是两个参数同时改变符号）：

$$\begin{array}{c} \xrightarrow{\mu} \\ \begin{array}{ccc} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{array} \\ \downarrow \nu \end{array}$$

183

即

$$\begin{aligned} F(\mu, \nu) &= 1 && \text{当 } (\mu, \nu) = (0, -1), (-1, 0), (1, 0), (0, 1) \text{ 时} \\ F(\mu, \nu) &= 0 && \text{当 } (\mu, \nu) = (-1, -1), (1, -1), (0, 0), (-1, 1), (1, 1) \text{ 时} \end{aligned}$$

形式化描述这个问题以便用线性规划的方法来求解。将每个所涉及的阵列的维数和元素表述清楚。（建议在傅里叶空间中至多用9个约束点）

由于对称性，有

$$\begin{aligned} h(-1, -1) &= h(1, 1) \\ h(-1, 0) &= h(1, 0) \\ h(0, -1) &= h(0, 1) \\ h(-1, 1) &= h(1, -1) \end{aligned}$$

对于一个 3×3 的滤波器来说，显然式(5-65)中 $N=1$ ，则：

$$\begin{aligned} H(\mu, \nu) &= 2h(-1, 1) \cos(\nu - \mu) + 2h(0, 1) \cos \nu + 2h(1, 1) \cos(\mu + \nu) \\ &\quad + 2h(1, 0) \cos \mu + h(0, 0) \end{aligned}$$

$h(1, 1)$ 、 $h(1, 0)$ 、 $h(0, 1)$ 、 $h(1, -1)$ 和 $h(0, 0)$ 是需要确定的未知数。由于它们可以是负值或正值，这些是完全未知的。定义误差 $\delta \equiv \max_{\mu, \nu} |H(\mu, \nu) - F(\mu, \nu)|$ ，那么：

$$\delta \geq |H - F| \Rightarrow \begin{cases} \delta \geq H - F & \Rightarrow \delta - H \geq -F \\ \delta \geq -(H - F) & \Rightarrow \delta + H \geq F \end{cases}$$

在频率空间对于每个约束点都采用类似于上面的两个不等式，即，总共有18个不等式。非负的 δ 需要极小化，它是未知的。因此问题转化为：在下面的约束下，极小化 $Z = X_1$ ：

$$\mathbf{A}_{11}\mathbf{X}_1 + \mathbf{A}_{12}\mathbf{X}_2 \geq \mathbf{B}_1 \quad \text{共18个不等式}$$

$$\mathbf{X}_1 \geq 0 \quad \text{1个未知}$$

$$\mathbf{X}_2 \text{ 自由} \quad \text{5个未知}$$

其中 $\mathbf{X}_1 = \delta$

$$\mathbf{X}_2 = \begin{pmatrix} h(0, 0) \\ h(0, 1) \\ h(-1, 1) \\ h(1, 0) \\ h(1, 1) \end{pmatrix} \quad \mathbf{A}_{11} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

按下面的顺序考虑傅里叶空间上的点：

184

1	2	3
4	5	6
7	8	9
$(-1, 1)$	$(0, -1)$	$(1, -1)$
$(-1, 0)$	$(0, 0)$	$(1, 0)$
$(-1, -1)$	$(0, 1)$	$(1, 1)$

那么

$$\mathbf{B}_1 = \begin{pmatrix} 0 \\ 0 \\ -1 \\ 1 \\ 0 \\ 0 \\ -1 \\ 1 \\ 0 \\ 0 \\ -1 \\ 1 \\ 0 \\ 0 \\ -1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

每个元素是 $-F(\mu, \nu)$ 或 $F(\mu, \nu)$ 的值

$$A_{12} = \begin{pmatrix} -1 & -2\cos 1 & -2 & -2\cos 1 & -2\cos 2 \\ 1 & 2\cos 1 & 2 & 2\cos 1 & 2\cos 2 \\ -1 & -2\cos 1 & -2\cos 1 & -2 & -2\cos 1 \\ 1 & 2\cos 1 & 2\cos 1 & 2 & 2\cos 1 \\ -1 & -2\cos 1 & -2\cos 2 & -2\cos 1 & -2 \\ 1 & 2\cos 1 & 2\cos 2 & 2\cos 1 & 2 \\ -1 & -2 & -2\cos 1 & -2\cos 1 & -2\cos 1 \\ 1 & 2 & 2\cos 1 & 2\cos 1 & 2\cos 1 \\ -1 & -2 & -2 & -2 & -2 \\ 1 & 2 & 2 & 2 & 2 \\ -1 & -2 & -2\cos 1 & -2\cos 1 & -2\cos 1 \\ 1 & 2 & 2\cos 1 & 2\cos 1 & 2\cos 1 \\ -1 & -2\cos 1 & -2\cos 2 & -2\cos 1 & -2 \\ 1 & 2\cos 1 & 2\cos 2 & 2\cos 1 & 2 \\ -1 & -2\cos 1 & -2\cos 1 & -2 & -2\cos 1 \\ 1 & 2\cos 1 & 2\cos 1 & 2 & 2\cos 1 \\ -1 & -2\cos 1 & -2 & -2\cos 1 & -2\cos 2 \\ 1 & 2\cos 1 & 2 & 2\cos 1 & 2\cos 2 \end{pmatrix}$$

185

B5.4 式(5-65)的证明。

运用恒等式 $\cos \alpha \cos \beta = \frac{1}{2} [\cos(\alpha + \beta) + \cos(\alpha - \beta)]$ ，并且定义 $\frac{2\pi}{p} = A$ ，式(5-64)可以改写为：

$$H(\mu, \nu) = \frac{1}{2p^2} \sum_{k=-N}^N \sum_{l=-N}^N \sum_{n=-N}^N \sum_{m=-N}^N H(k, l) \{ \cos[An(\mu + k) + mA(l + \nu)] \\ + \cos[nA(k - \mu) + mA(l - \nu)] \}$$

定义：

$$A(\mu + k) \equiv x$$

$$A(l + \nu) \equiv y$$

$$A(k - \mu) \equiv u$$

$$A(l - \nu) \equiv v$$

那么

$$H(\mu, \nu) = \frac{1}{2p^2} \sum_{k=-N}^N \sum_{l=-N}^N H(k, l) \sum_{n=-N}^N \sum_{m=-N}^N \{ \cos(nx + my) + \cos(nu + mv) \} \quad (5-66)$$

首先考虑括弧里的第一项：

$$\sum_{n=-N}^N \sum_{m=-N}^N \cos(nx + my) = \sum_{n=-N}^N \sum_{m=-N}^N (\cos(nx) \cos(my) + \sin(my) \sin(nx))$$

$$= \sum_{n=-N}^N \cos(nx) \sum_{m=-N}^N \cos(my) \\ + \sum_{m=-N}^N \sin(my) \sum_{n=-N}^N \sin(nx)$$

由于sine是奇函数, 因此 $\sin(my)$ 在一个对称的区域上求和后得到是0。这样上式的第二项可以消掉。另一方面, $\cos(nx)$ 是偶函数, 上式可写为:

$$\sum_{n=-N}^N \sum_{m=-N}^N \cos(nx + my) = \left(1 + 2 \sum_{n=1}^N \cos(nx)\right) \left(1 + 2 \sum_{m=1}^N \cos(my)\right)$$

186

利用下面的恒等式代入上式:

$$\sum_{k=1}^n \cos(kx) = \cos\left(\frac{n+1}{2}x\right) \sin \frac{nx}{2} \frac{1}{\sin \frac{x}{2}}$$

可以得到:

$$\sum_{n=-N}^N \sum_{m=-N}^N \cos(nx + my) = \left(1 + \frac{2 \cos\left(\frac{N+1}{2}x\right) \sin \frac{Nx}{2}}{\sin \frac{x}{2}}\right) \left(1 + \frac{2 \cos\left(\frac{N+1}{2}y\right) \sin \frac{Ny}{2}}{\sin \frac{y}{2}}\right)$$

同样, 运用恒等式: $2\cos\alpha\sin\beta = \sin(\alpha + \beta) - \sin(\alpha - \beta)$ 。可以得到:

$$\begin{aligned} \sum_{n=-N}^N \sum_{m=-N}^N \cos(nx + my) &= \left(1 + \frac{\sin\left(\frac{2N+1}{2}x\right) - \sin \frac{x}{2}}{\sin \frac{x}{2}}\right) \left(1 + \frac{\sin\left(\frac{2N+1}{2}y\right) - \sin \frac{y}{2}}{\sin \frac{y}{2}}\right) \\ &= \frac{\sin\left(\frac{2N+1}{2}x\right) \sin\left(\frac{2N+1}{2}y\right)}{\sin \frac{x}{2} \sin \frac{y}{2}} \end{aligned} \quad (5-67)$$

定义:

$$\phi(\alpha, \beta) \equiv \frac{\sin[\pi(\alpha + \beta)]}{\sin[\frac{2\pi}{p}(\alpha + \beta)]} \quad (5-68)$$

由 p , x 和 A 的定义, 有: $\frac{2N+1}{2}x = \frac{2N+1}{2}A(\mu + k) = \frac{p}{2} \frac{2\pi}{p}(\mu + \nu) = \pi(\mu + \nu)$ 。对于

$\frac{2N+1}{2}y$ 有类似的结果。因此式(5-67)可以写为:

$$\sum_{n=-N}^N \sum_{m=-N}^N \cos(nx + my) = \phi(\mu, k)\phi(l, \nu) \quad (5-69)$$

用类似的方法可以得到:

$$\sum_{n=-N}^N \sum_{m=-N}^N \cos(nu + mv) = \phi(k, -\mu)\phi(l, -\nu)$$

那么式(5-66)可以写为:

$$H(\mu, \nu) = \frac{1}{2p^2} \sum_{k=-N}^N \sum_{l=-N}^N H(k, l) \{ \phi(\mu, k) \phi(l, \nu) + \phi(k, -\mu) \phi(l, -\nu) \} \quad (5-70)$$

定义:

$$\Phi(k, l) \equiv \phi(\mu, k) \phi(l, \nu) + \phi(k, -\mu) \phi(l, -\nu) \quad (5-71)$$

由 ϕ 的定义(式(5-68)), 显然有:

$$\Phi(k, l) = \Phi(-k, -l)$$

同样有 $H(k, l) = H(-k, -l)$ 。那么式(5-70)可以进一步简化为:

$$\begin{aligned} H(\mu, \nu) &= \frac{1}{2p^2} \sum_{k=-N}^N \left[\sum_{l=-N}^{-1} H(k, l) \Phi(k, l) + \sum_{l=1}^N H(k, l) \Phi(k, l) + H(k, 0) \Phi(k, 0) \right] \\ &= \frac{1}{2p^2} \sum_{k=-N}^N \left[\sum_{\tilde{l}=1}^N H(k, -\tilde{l}) \Phi(k, -\tilde{l}) + \sum_{l=1}^N H(k, l) \Phi(k, l) + H(k, 0) \Phi(k, 0) \right] \end{aligned}$$

用类似的方式, 把 k 求和的式子拆成0项、正的部分和负的部分, 定义新的变量 $\tilde{k} = -k$ 来对 k 为负的部分求和:

$$\begin{aligned} H(\mu, \nu) &= \frac{1}{2p^2} \left[\underbrace{\sum_{\tilde{k}=1}^N \sum_{l=1}^N H(-\tilde{k}, -l) \Phi(-\tilde{k}, -l)}_{\text{负部分}} + \sum_{\tilde{k}=1}^N \sum_{l=1}^N H(-\tilde{k}, l) \Phi(-\tilde{k}, l) \right. \\ &\quad + \underbrace{\sum_{\tilde{k}=1}^N H(-\tilde{k}, 0) \Phi(-\tilde{k}, 0)}_{\text{0项}} + \underbrace{\sum_{k=1}^N \sum_{l=1}^N H(k, -l) \Phi(k, -l)}_{\text{正部分}} \\ &\quad + \underbrace{\sum_{k=1}^N \sum_{l=1}^N H(k, l) \Phi(k, l)}_{\text{正部分}} \\ &\quad + \underbrace{\sum_{k=1}^N H(k, 0) \Phi(k, 0)}_{\text{0项}} + \underbrace{\sum_{\tilde{l}=1}^N H(0, -\tilde{l}) \Phi(0, -\tilde{l})}_{\text{0项}} \\ &\quad \left. + \underbrace{\sum_{\tilde{l}=1}^N H(0, \tilde{l}) \Phi(0, \tilde{l}) + H(0, 0) \Phi(0, 0)}_{\text{0项}} \right] \end{aligned}$$

由于函数 $\Phi(k, l)$ 和 $H(k, l)$ 的对称性, 上式中有同样下划线的部分是相同的, 因此将它们合并得到:

$$H(\mu, \nu) = \frac{1}{2p^2} \left[2 \sum_{k=1}^N \sum_{l=1}^N H(k, l) \Phi(k, l) + 2 \sum_{k=1}^N \sum_{l=1}^N H(k, -l) \Phi(k, -l) \right]$$

$$+ 2 \sum_{k=1}^N H(k, 0) \Phi(k, 0) + 2 \sum_{l=1}^N H(0, l) \Phi(0, l) + H(0, 0) \Phi(0, 0) \Big] \quad (5-72)$$

上式右边的前三项可以合并成一项，因为这三项完全包括了对 l 求和的正的部分、负的部分和0项，得到：

$$H(\mu, \nu) = \frac{1}{2p^2} \left[2 \sum_{k=1}^N \sum_{l=-N}^N H(k, l) \Phi(k, l) + 2 \sum_{l=1}^N H(0, l) \Phi(0, l) + H(0, 0) \Phi(0, 0) \right]$$

最后，代换上式的 p ，得到：

$$H(\mu, \nu) = \frac{1}{(2N+1)^2} \left[\frac{1}{2} H(0, 0) \Phi(0, 0) + \sum_{l=1}^N H(0, l) \Phi(0, l) + \sum_{k=1}^N \sum_{l=-N}^N H(k, l) \Phi(k, l) \right] \quad (5-73)$$

5.36 当仅在频域设计滤波器时要怎样做

需要找到 $H(k, l)$, $k = 0, 1, \dots, N$, $l = 0, 1, \dots, N$, 的系数最优值，使得在固定数目的约束点 (μ_i, ν_i) 用 $H(\mu, \nu)$ 逼近 $F(\mu, \nu)$ 的误差达到最小。

实际应用的时候，通常的做法是固定 $H(k, l)$ 的一些值，然后对余下的值进行最优化选择。例如，在低通滤波器情况下，对于我们想要保留的频率，令 $H(k, l) = 1$ ，而对不需要的频率，令 $H(k, l) = 0$ 。然后余下的过度频率 $H(k, l)$ 将被确定（如图5-7所示）。

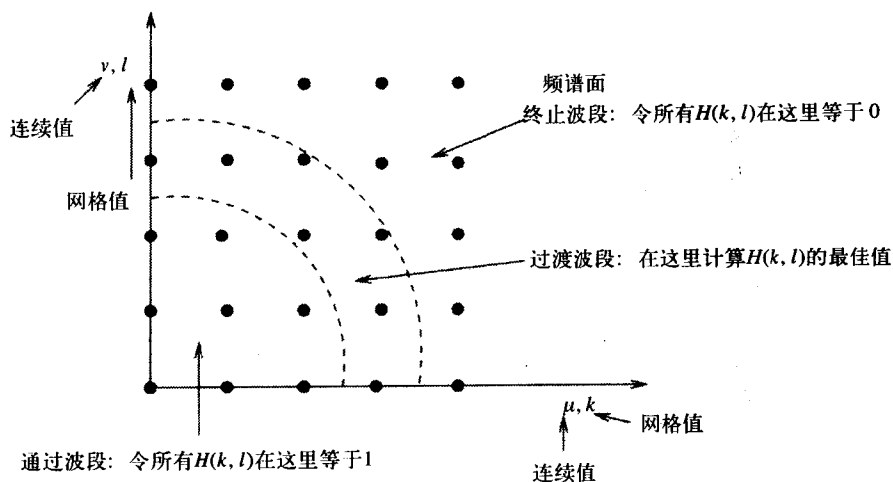


图 5-7 在频域中固定某些滤波器的值，而让其他值自由

5.37 如何求解未知值 $H(k, l)$

类似于式(5-50), 可以用线性规划的方法求解式(5-65)。

5.38 根据Chebyshev准则频率采样方法是否能得到最优解

不能, 因为 $H(k, l)$ 的某些值是固定的, 所以并没有对全部 $H(k, l)$ 的值求最优化。

• 例5.5 在频域上用形式化描述滤波器设计问题以便可以用线性规划包求解。

比较式(5-65)和式(5-50), 可以看出该问题类似于在实域上用线性规划的方法确定滤波器的值: 在实域的情形下, 试图确定 $h(0, 0), h(1, 0), \dots, h(N, 0), \dots, h(-N, m), h(-N+1, m), \dots, h(N, m)$, 当 $m = 1, 2, \dots, N$, 而现在在频域的情形下, 是要确定 $H(0, 0), H(0, 1), \dots, H(0, N), H(k, -N), H(k, -N+1), \dots, H(k, N)$, 当 $k = 1, 2, \dots, N$ 。

在实域的情形下, 式(5-50)中出现了函数 $\cos(\mu n + \nu m)$ 。现在有对应的函数 $\Phi(\mu, \nu)$ 。同样, $\Phi(\mu, \nu)$ 对于两个参数也是一个偶函数, 因此, 它在相关问题的求解中的作用跟余弦函数类似。由此, 问题可以表述为:

最小化:

$$z = x_{11}$$

约束条件为:

$$A_{11}x_{11} + A_{12}x_2 \geq B_1$$

其中:

$$x_{11} = \delta$$

并且

$$x_2 = \begin{pmatrix} H(0,0) \\ H(1,0) \\ H(2,0) \\ \vdots \\ H(N,0) \\ H(-N,1) \\ \vdots \\ H(N,1) \\ H(-N,2) \\ \vdots \\ H(N,2) \\ \vdots \\ H(-N,N) \\ \vdots \\ H(N,N) \end{pmatrix} \leftarrow \begin{matrix} (2N^2 + 2N + 1) \times 1 \\ \text{矩阵} \end{matrix} \quad B_1 = \begin{pmatrix} F(\mu_1, \nu_1) \\ -F(\mu_1, \nu_1) \\ F(\mu_2, \nu_2) \\ -F(\mu_2, \nu_2) \\ \vdots \\ F(\mu_{p_1}, \nu_{p_1}) \\ -F(\mu_{p_1}, \nu_{p_1}) \end{pmatrix} \leftarrow \begin{matrix} 2p_1 \times 1 \\ \text{矩阵} \end{matrix}$$

$$A_{11} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \leftarrow \begin{matrix} 2p_1 \times 1 \\ \text{矩阵} \end{matrix}$$

A_{12} 是一个 $2p_1 \times (2N^2 + 2N + 1)$ 的矩阵, 其中的元素可以通过把式(5-65)代入不等式约束中得到。它可以写为:

$$A_{12} = \begin{pmatrix} 1 & 2\Phi(\mu_1, 0) & 2\Phi(2\mu_1, 0) & \dots & 2\Phi(N\mu_1, 0) & 2\Phi(-N\mu_1, \nu_1) \\ -1 & -2\Phi(\mu_1, 0) & -2\Phi(2\mu_1, 0) & \dots & -2\Phi(N\mu_1, 0) & -2\Phi(-N\mu_1, \nu_1) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 2\Phi(\mu_{p_1}, 0) & 2\Phi(2\mu_{p_1}, 0) & \dots & 2\Phi(N\mu_{p_1}, 0) & 2\Phi(-N\mu_{p_1}, \nu_{p_1}) \\ -1 & -2\Phi(\mu_{p_1}, 0) & -2\Phi(2\mu_{p_1}, 0) & \dots & -2\Phi(N\mu_{p_1}, 0) & -2\Phi(-N\mu_{p_1}, \nu_{p_1}) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & 2\Phi(N\mu_1, \nu_1) & \dots & 2\Phi(-N\mu_1, N\nu_1) & \dots & 2\Phi(N\mu_1, N\nu_1) \\ \dots & -2\Phi(N\mu_1, \nu_1) & \dots & -2\Phi(-N\mu_1, N\nu_1) & \dots & -2\Phi(N\mu_1, N\nu_1) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & 2\Phi(N\mu_{p_1}, \nu_{p_1}) & \dots & 2\Phi(-N\mu_{p_1}, N\nu_{p_1}) & \dots & 2\Phi(N\mu_{p_1}, N\nu_{p_1}) \\ \dots & -2\Phi(N\mu_{p_1}, \nu_{p_1}) & \dots & -2\Phi(-N\mu_{p_1}, N\nu_{p_1}) & \dots & -2\Phi(N\mu_{p_1}, N\nu_{p_1}) \end{pmatrix}$$

191

5.39 本章要点

如果准确指定一个滤波器频率响应在有限的区域之外为0, 我们便获得了一个实域上的无限滤波器。一个数字化的无限滤波器可以借助 z 变换表达成递归滤波器。二维 z 变换比一维 z 变换更难求逆。还有, 递归滤波器要满足特定的准则以便稳定。二维数字滤波器理论、它们的稳定性准则以及它们的设计问题已经超出了本书的范围。

为了避免用无限脉冲响应滤波器, 可以用逼近的方法。首先确定所需滤波器频率响应, 然后试图去找有限的脉冲响应滤波器, 使得它能在要求的频率上有尽可能好的逼近。

有时, 能在一些频率上准确获得所需的频率响应是很重要的, 而对于其他的频率不必太关心。因此, 在特定的频率上确定滤波器的频率响应, 然后用逼近的方法确定该滤波器在其他频率上的值。

在滤波器设计的逼近的准则中, 用的是Chebyshev范数而不是最小均方误差。这是因为我们更关心的是在逼近过程中每个频率上的误差, 而不是全部频率不均匀分布所产生的整体误差。

192

第6章 图像复原

6.1 什么是图像复原

图像复原是指使用客观标准和图像本身的先验知识来改善给定图像的过程。

6.2 图像增强和图像复原之间的区别是什么

图像增强是指使用主观标准改善图像，而图像复原是指采用与图像受损过程相反的过程来恢复图像，这里使用的是客观标准。

6.3 为什么图像需要复原

图像个别像素灰度值的改变可能导致图像的退化，而个别像素偏离正确位置也可能引起图像的畸变。后者正是几何复原 (geometric restoration) 的研究主题。

几何复原又称为图像对准 (image registration)，因为它借助寻找同一区域在不同视角下的两幅图像的对应点对处理。图像对准在遥感应用中是非常重要的，例如航空照片要和地图进行校准，或者同一区域的两幅航空图片互相校准。

6.4 几何畸变是如何产生的

透镜或者采集图像过程中传感器的不规则移动都可能造成图像的几何畸变 (geometric distortion)。在第一种情况下，畸变如图6-1所示，这种变形看上去是规则的。而第二种情况产生在诸如用行扫描相机从空中拍摄地球表面照片的时候。当飞机摇摆时，拍摄的图像就会产生非均匀的扭曲，像素从真实位置移动的距离会达到像素间距的4~5倍。

193

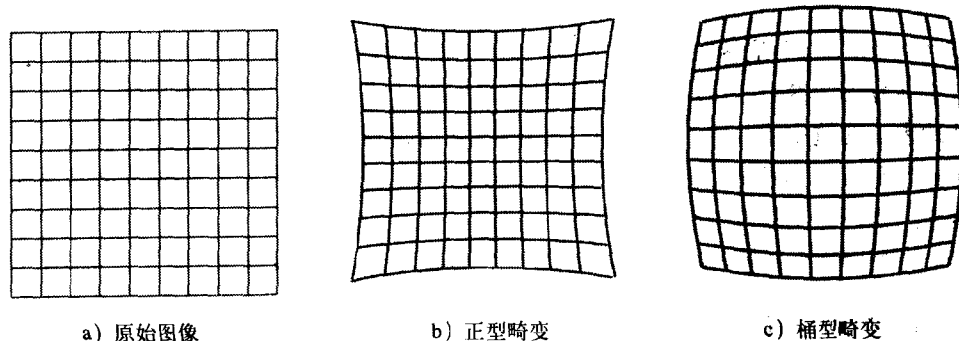


图6-1 由透镜产生的几何畸变的例子

6.5 几何畸变的图像如何被复原

我们先创建一个与畸变图像大小相同的空数字矩阵，然后用这个矩阵存放校正后的图像。我们的目标是给这个矩阵中的每个元素赋予灰度值。可以通过两步运算来实现：第一步插入灰度值；第二步空间转换。

6.6 如何实现空间变换

假设像素的真实位置是 (x, y) ，畸变后的位置是 (\hat{x}, \hat{y}) (见图6-2)。一般地，存在一个使一组坐标变换到另一组坐标的变换，即：

$$\hat{x} = O_x(x, y) \quad (6-1)$$

$$\hat{y} = O_y(x, y) \quad (6-2)$$

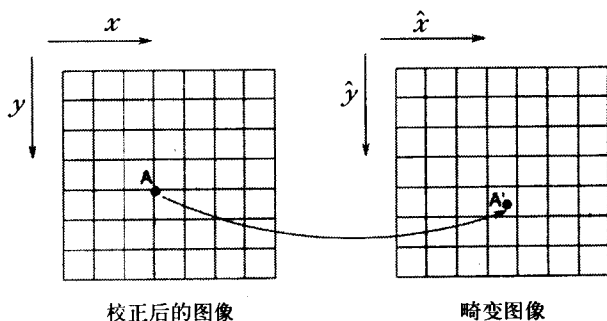


图6-2 图中的像素对应于网格的节点，校正后的图像的网格中的像素A对应于原图像中的像素A'

首先，我们必须找到校正后图像中的每个像素在畸变图像中对应的坐标位置。这里，我们经常会做一些假设，例如，我们可以假设上述变换具有下列形式：

$$\hat{x} = c_1x + c_2y + c_3xy + c_4 \quad (6-3)$$

$$\hat{y} = c_5x + c_6y + c_7xy + c_8 \quad (6-4)$$

其中 c_1, c_2, \dots, c_8 是参数。另外，我们也可以假设更一般的形式，上述等式的右边可以出现 x 和 y 的平方项。参数 c_1, c_2, \dots, c_8 的值是由已知点（称为结点）的变换来确定的。例如，在拍摄地球表面的航空照片中，总有一些已知确切位置的地标。地球表面分布着许多类似的点。我们可以利用四个这样的点来确定上述八个参数的值，并假设具有这些参数值的变换公式在这四个结点所定义的四边形中都成立。

然后，利用这个变换，就可以找到校正后的图像中点A在畸变图像中的对应点A'的位置了。

6.7 为什么灰度插值是必要的

尽管点A在 (x, y) 空间中的坐标是整数，但点A'的坐标似乎并不是整数。这就意味着我

们实际上不知道 A' 位置的精确灰度值,需要进行灰度插值处理。 A' 位置的灰度值可以由 (\hat{x}, \hat{y}) 空间中四个最近邻像素的值来估算,例如,可以采用双线性插值的方法。假设在每个小方格内灰度值都是位置坐标的简单函数:

$$g(\hat{x}, \hat{y}) = \alpha \hat{x} + \beta \hat{y} + \gamma \hat{x}\hat{y} + \delta$$

其中 α, \dots, δ 是一些参数。将四个角的像素值代入上式,可推导出 α, β, γ 和 δ ,然后利用这些值来计算 A' 位置的 $g(\hat{x}, \hat{y})$ 。

图6-3是畸变图像中点 A' 的邻域的放大图,离 A' 点四个最近的像素都具有整数坐标。

当然也可以采用更简单或者更复杂的插值方法。例如,最简单的方法就是最近邻插值法, A' 点的灰度值可以取其最近邻像素的灰度值。而更复杂的方法则是通过 A' 附近更大的像素块来构造更高阶的表面方程,然后通过解这个方程求出 A' 的值。

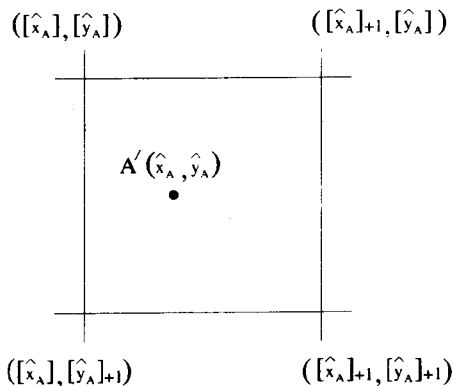
194
195

图6-3 点 A' 在非整数位置,它周围的四个像素都在整数位置并且灰度值已知 ($[x]$ 是指 x 的整数部分)

- 例6.1 下图中,右边的网格图是几何畸变图像,且将左边参考图像中点 A, B, C, D 的对应点标记为结点。左图中各项表示坐标位置。假设矩形 $ABCD$ 的畸变可以由双线性插值来建模,并且其内部像素的灰度值也可以由双线性插值来建模,那么我们就可以计算出参考图像中像素位置 $(2, 2)$ 的灰度值。

A			B		A
$(0, 0)$	(1, 0)	(2, 0)	$(3, 0)$	(4, 0)	4 3 2 5 1
(0, 1)	(1, 1)	(2, 1)	(3, 1)	(4, 1)	3 3 4 2B 4
(0, 2)	(1, 2)	(2, 2)	(3, 2)	(4, 2)	0 5 5 3 3
$C(0, 3)$	(1, 3)	(2, 3)	$(3, 3)D$	(4, 3)	0 4C 2 1 2
(0, 4)	(1, 4)	(2, 4)	(3, 4)	(4, 4)	2 1 0 3 5D

假设像素在畸变图像中的位置 (\hat{x}, \hat{y}) 对应参考图像中的位置 (x, y) :

$$\begin{aligned}\hat{x} &= c_1x + c_2y + c_3xy + c_4 \\ \hat{y} &= c_5x + c_6y + c_7xy + c_8\end{aligned}$$

196

下面我们将给出两幅图像中使用的四个结点的相对坐标:

	畸变图像的对应坐标 (\hat{x}, \hat{y})	参考图像的坐标 (x, y)
像素A	(0, 0)	(0, 0)
像素B	(3, 1)	(3, 0)
像素C	(1, 3)	(0, 3)
像素D	(4, 4)	(3, 3)

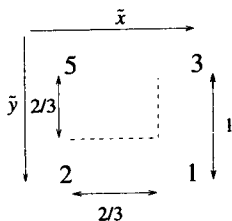
利用这些, 可以计算参数 c_1, c_2, \dots, c_8 的值:

$$\begin{aligned}
 &\text{像素A:} \quad \begin{matrix} c_4 = 0 \\ c_8 = 0 \end{matrix} \\
 &\text{像素B:} \quad \begin{matrix} 3 = 3c_1 \\ 1 = 3c_5 \end{matrix} \Rightarrow \begin{cases} c_1 = 1 \\ c_5 = \frac{1}{3} \end{cases} \\
 &\text{像素C:} \quad \begin{matrix} 1 = 3c_2 \\ 3 = 3c_6 \end{matrix} \Rightarrow \begin{cases} c_2 = \frac{1}{3} \\ c_6 = 1 \end{cases} \\
 &\text{像素D:} \quad \begin{matrix} 4 = 3 + 3 \times \frac{1}{3} + 9c_3 \\ 4 = 3 \times \frac{1}{3} + 3 + 9c_7 \end{matrix} \Rightarrow \begin{cases} c_3 = 0 \\ c_7 = 0 \end{cases}
 \end{aligned}$$

因此, 矩形ABCD中的任何像素在畸变后的对应坐标都可以由下式求出:

$$\hat{x} = x + \frac{y}{3}, \quad \hat{y} = \frac{x}{3} + y$$

对于 $x = y = 2$, 可以得到 $\hat{x} = 2 + \frac{2}{3}$, $\hat{y} = \frac{2}{3} + 2$ 。因此, 像素(2, 2)在畸变图像中对应坐标是 $(2\frac{2}{3}, 2\frac{2}{3})$, 这个位置位于畸变图像的像素之间, 具体地说它处于具有下述灰度值的像素之间:



我们定义一个局部坐标系 (\tilde{x}, \tilde{y}) , 使得系统中左上角像素对应坐标(0, 0), 右上角像素对应坐标(1, 0), 左下角和右下角像素分别对应坐标(0, 1)和(1, 1)。假设这四个像素间的像素灰度值可以由这四个角上的像素灰度值通过双线性插值计算得到, 则有:

$$g(\tilde{x}, \tilde{y}) = \alpha \tilde{x} + \beta \tilde{y} + \gamma \tilde{x}\tilde{y} + \delta$$

将四个邻近像素代入, 得:

$$\begin{aligned}
 5 &= \alpha \cdot 0 + \beta \cdot 0 + \gamma \cdot 0 + \delta \Rightarrow \delta = 5 \\
 3 &= \alpha \cdot 1 + \beta \cdot 0 + \gamma \cdot 0 + 5 \Rightarrow \alpha = -2
 \end{aligned}$$

$$\begin{aligned} 2 &= (-2) \cdot 0 + \beta \cdot 1 + \gamma \cdot 0 + 5 \Rightarrow \beta = -3 \\ 1 &= (-2) \cdot 1 + (-3) \cdot 1 + \gamma \cdot 1 \cdot 1 + 5 \Rightarrow \gamma = 1 \end{aligned}$$

故有

$$g(\tilde{x}, \tilde{y}) = -2\tilde{x} - 3\tilde{y} + \tilde{x}\tilde{y} + 5$$

将 $\tilde{x} = \frac{2}{3}$ 和 $\tilde{y} = \frac{2}{3}$ 代入, 可得:

$$g = -2 \times \frac{2}{3} - 3 \times \frac{2}{3} + \frac{2}{3} \times \frac{2}{3} + 5 = -\frac{4}{3} - \frac{6}{3} + \frac{4}{9} + 5 = 2\frac{1}{9}$$

因此, 点(2, 2)处的灰度值为 $2\frac{1}{9}$, 四舍五入后得到2。如果采用的是最近邻插值法, 那么此处的灰度值将是1, 因为它最近邻的像素是右下角的像素点。

6.8 退化图像是如何依赖非退化图像和线性退化过程的点扩展函数的

假设对图像造成破坏性的影响是线性的, 输出图像 $g(\alpha, \beta)$ 可由输入图像 $f(x, y)$ 表示如下:

$$g(\alpha, \beta) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) h(x, y, \alpha, \beta) dx dy \quad (6-5)$$

其中, $h(x, y, \alpha, \beta)$ 是表示退化影响的点扩散函数。

6.9 退化图像是如何依赖于非退化图像和线性移不变退化过程的点扩展函数的

如果我们有

$$h(x, y, \alpha, \beta) = h(\alpha - x, \beta - y) \quad (6-6)$$

则称 h 为移不变点扩散函数。这就意味着图像中某点的输出只依赖于该点与图像中其他点的相对位置, 而与该点的实际位置无关。

可得

$$g(\alpha, \beta) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) h(\alpha - x, \beta - y) dx dy \quad (6-7) \quad \boxed{198}$$

可以看出式(6-7)实际上是非退化图像 $f(x, y)$ 和点扩散函数的卷积, 可以写出其傅里叶变换形式:

$$\hat{G}(u, v) = \hat{F}(u, v) \hat{H}(u, v) \quad (6-8)$$

其中, \hat{G} 、 \hat{F} 和 \hat{H} 分别为函数 f 、 g 和 h 的傅里叶变换。

6.10 对于离散图像式(6-5)的形式是怎样的

$$g(i, j) = \sum_{k=1}^N \sum_{l=1}^N f(k, l) h(k, l, i, j) \quad (6-9)$$

式(6-9)可以写成矩阵形式 (参见式(1-25)):

$$\mathbf{g} = \mathbf{H}\mathbf{f} \quad (6-10)$$

6.11 图像复原的问题是什么

图像复原的问题是: 给定退化图像 g , 要将它恢复成初始未退化图像 f 。

6.12 如何解决图像复原的问题

如果有退化过程的点扩散函数或者其傅里叶变换 (即传递函数) 的先验知识, 那么就可以解决图像复原的问题。

6.13 如何获得退化过程的传递函数 $\hat{H}(u, v)$ 的信息

1) 由引起退化的物理过程中的知识获得。例如, 当退化是由衍射产生时, 可以计算出 $H(u, v)$ 。同样地, 如果退化是由大气湍流或者运动产生的, 也可以建模并计算 $H(u, v)$ 。

2) 可以从图像本身提取 $\hat{H}(u, v)$ 或 $h(\alpha-x, \beta-y)$ 的信息。即, 忽略隐含的物理过程所产生的实际状态, 仅从该过程对已知对象的图像产生的影响入手来获取信息。

199

- 例6.2 当摄录特定静态场景时, 相机沿平行于图像平面 (x, y) 的平面移动, 这等价于静态场景
- 沿 x, y 方向分别移动距离 $x_0(t), y_0(t)$, 其中 $x_0(t)$ 和 $y_0(t)$ 都是时间 t 的函数。相机延时从 $t=0$ 到 $t=T$ (T 是常数)。于是, 可以得出像素位置 (x, y) 处的亮度值的表达式, 它以场景亮度函数 $f(x, y)$ 的形式表达出来。

底片任意点的曝光时间都将是 T , 于是可以得到模糊图像:

$$g(x, y) = \int_0^T f(x - x_0(t), y - y_0(t)) dt \quad (6-11)$$

例6.3 例6.2给出了传递函数, 利用传递函数我们可以对由相机移动产生的图像退化过程建模。假设退化过程是移不变线性点扩散函数。

考虑例6.2中定义的函数 $g(x, y)$ 的傅里叶变换:

$$\hat{G}(u, v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} g(x, y) e^{-2\pi j(ux+vy)} dx dy \quad (6-12)$$

将式(6-11)代入式(6-12), 得:

$$\hat{G}(u, v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_0^T f(x - x_0(t), y - y_0(t)) dt e^{-2\pi j(ux + vy)} dx dy \quad (6-13)$$

交换积分顺序, 则有:

$$\hat{G}(u, v) = \int_0^T \underbrace{\left\{ \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x - x_0(t), y - y_0(t)) e^{-2\pi j(ux + vy)} dx dy \right\}}_{\text{这是平移函数分别沿 } x, y \text{ 方向平移 } x_0, y_0 \text{ 的傅里叶变换}} dt \quad (6-14) \quad \boxed{200}$$

由上述(参见式(2-67))可知, 平移函数的傅里叶变换和非平移函数的傅里叶变换之间的关系可由下式给出:

$$(\text{平移函数的傅里叶变换}) = (\text{非平移函数的傅里叶变换}) e^{-2\pi j(ux_0 + vy_0)}$$

因此可得:

$$\hat{G}(u, v) = \int_0^T \hat{F}(u, v) e^{-2\pi j(ux_0 + vy_0)} dt$$

其中, $\hat{F}(u, v)$ 是场景亮度函数 $f(x, y)$ 的傅里叶变换, 即未经模糊的图像。 $\hat{F}(u, v)$ 与时间无关, 所以可以从积分符号中提出:

$$\hat{G}(u, v) = \hat{F}(u, v) \int_0^T e^{-2\pi j(ux_0(t) + vy_0(t))} dt$$

与式(6-8)相比较, 可得:

$$\hat{H}(u, v) = \int_0^T e^{-2\pi j(ux_0(t) + vy_0(t))} dt \quad (6-15)$$

例6.4 设例6.2中的运动只沿 x 方向且保持常速度 $\frac{\alpha}{T}$, 则 $y_0(t) = 0$, $x_0(t) = \frac{\alpha t}{T}$ 。下面来计算产生运动模糊的传递函数。

将 $y_0(t)$ 和 $x_0(t)$ 代入例6.3的结果(式(6-15)中), 可得:

$$\begin{aligned} \hat{H}(u, v) &= \int_0^T e^{-2\pi j u \frac{\alpha t}{T}} dt = \frac{e^{-2\pi j u \frac{\alpha t}{T}}}{-2\pi j u \frac{\alpha}{T}} \bigg|_0^T = -\frac{T}{2\pi j u \alpha} [e^{-2\pi j u \alpha} - 1] \\ &= \frac{T}{2\pi j u \alpha} [1 - e^{-2\pi j u \alpha}] = \frac{T e^{-\pi j u \alpha}}{2\pi j u \alpha} [e^{\pi j u \alpha} - e^{-\pi j u \alpha}] \Rightarrow \\ \hat{H}(u, v) &= \frac{T e^{-\pi j u \alpha} 2j \sin(\pi u \alpha)}{2\pi j u \alpha} = T \frac{\sin(\pi u \alpha)}{\pi u \alpha} e^{-j\pi u \alpha} \end{aligned} \quad (6-16) \quad \boxed{201}$$

例6.5(B) 假设相机延时的时间间隔为 T , 由于相机的移动使我们感觉好像是场景中的物体由零位移处开始, 以初速度 s_0 沿 y 轴正向移动且有固定加速度 2α 。下面来推导此例中退化过程的传递函数。

此例中 $x_0(t)=0$ 且

$$\frac{d^2 y_0}{dt^2} = 2\alpha \Rightarrow \frac{dy_0}{dt} = 2\alpha t + b \Rightarrow y_0(t) = \alpha t^2 + bt + c$$

其中, α 为固定加速度的一半, b 和 c 为整型常量。且有下列初始条件:

$$t = 0 \quad \text{零位移 即 } c = 0$$

$$t = 0 \quad \text{位移速度} = s_0 \Rightarrow b = s_0$$

因此:

$$y_0(t) = \alpha t^2 + s_0 t$$

将 $x_0(t)$ 和 $y_0(t)$ 代入式(6-15), 则:

$$\begin{aligned} H(u, v) &= \int_0^T e^{-2\pi j v (\alpha t^2 + s_0 t)} dt \\ &= \int_0^T \cos [2\pi v \alpha t^2 + 2\pi v s_0 t] dt - j \int_0^T \sin [2\pi v \alpha t^2 + 2\pi v s_0 t] dt \end{aligned}$$

可以使用下列公式:

$$\begin{aligned} \int \cos(ax^2 + bx + c) dx &= \sqrt{\frac{\pi}{2a}} \left\{ \cos \frac{ac - b^2}{a} C \left(\frac{ax + b}{\sqrt{a}} \right) - \sin \frac{ac - b^2}{a} S \left(\frac{ax + b}{\sqrt{a}} \right) \right\} \\ \int \sin(ax^2 + bx + c) dx &= \sqrt{\frac{\pi}{2a}} \left\{ \cos \frac{ac - b^2}{a} S \left(\frac{ax + b}{\sqrt{a}} \right) + \sin \frac{ac - b^2}{a} C \left(\frac{ax + b}{\sqrt{a}} \right) \right\} \end{aligned}$$

其中 $S(x)$ 和 $C(x)$ 分别为:

$$\begin{aligned} S(x) &\equiv \sqrt{\frac{2}{\pi}} \int_0^x \sin t^2 dt \\ C(x) &\equiv \sqrt{\frac{2}{\pi}} \int_0^x \cos t^2 dt \end{aligned}$$

上述公式又被称作Fresnel积分。

使用上述公式, 其中:

$$\begin{aligned} a &\rightarrow 2\pi v \alpha \\ b &\rightarrow 2\pi v s_0 \\ c &\rightarrow 0 \end{aligned}$$

可得:

$$H(u, v) = \frac{1}{2\sqrt{v\alpha}} \left\{ \cos \frac{2\pi v s_0^2}{\alpha} C \left(\frac{\sqrt{2\pi v} (\alpha t + s_0)}{\sqrt{\alpha}} \right) \right.$$

$$\begin{aligned}
& -\sin\left(-\frac{2\pi v s_0^2}{\alpha}\right) S\left(\sqrt{2\pi v} \frac{\alpha t + s_0}{\sqrt{\alpha}}\right) - j \cos \frac{2\pi v s_0^2}{\alpha} S\left(\sqrt{2\pi v} \frac{\alpha t + s_0}{\sqrt{\alpha}}\right) \\
& -j \sin\left(-\frac{2\pi v s_0^2}{\alpha}\right) C\left(\sqrt{2\pi v} \frac{\alpha t + s_0}{\sqrt{\alpha}}\right) \Bigg\} \Bigg|_0^T \\
= & \frac{1}{2\sqrt{v\alpha}} \left\{ \cos \frac{2\pi v s_0^2}{\alpha} \left[C\left(\sqrt{\frac{2\pi v}{\alpha}}(\alpha T + s_0)\right) - j S\left(\sqrt{\frac{2\pi v}{\alpha}}(\alpha T + s_0)\right) \right] \right. \\
& + \sin \frac{2\pi v s_0^2}{\alpha} \left[S\left(\sqrt{\frac{2\pi v}{\alpha}}(\alpha T + s_0)\right) + j C\left(\sqrt{\frac{2\pi v}{\alpha}}(\alpha T + s_0)\right) \right] \\
& - \cos \frac{2\pi v s_0^2}{\alpha} \left[C\left(\sqrt{\frac{2\pi v}{\alpha}}s_0\right) - j S\left(\sqrt{\frac{2\pi v}{\alpha}}s_0\right) \right] \\
& \left. - \sin \frac{2\pi v s_0^2}{\alpha} \left[S\left(\sqrt{\frac{2\pi v}{\alpha}}s_0\right) + j C\left(\sqrt{\frac{2\pi v}{\alpha}}s_0\right) \right] \right\} \quad (6-17)
\end{aligned}$$

例6.6(B) 如果快门长时间打开, 并且平移的初始速度可以忽略, 那么例6.5中的传递函数又是什么形式呢?

已知例6.5结果中出现的函数 $S(x)$ 和 $C(x)$ 的极限形式如下所示:

203

$$\begin{aligned}
\lim_{x \rightarrow \infty} S(x) &= \frac{1}{2} \\
\lim_{x \rightarrow \infty} C(x) &= \frac{1}{2} \\
\lim_{x \rightarrow 0} S(x) &= 0 \\
\lim_{x \rightarrow 0} C(x) &= 0
\end{aligned}$$

因此, 对 $s_0 \approx 0$ 和 $T \rightarrow \infty$, 可得:

$$\begin{aligned}
C\left(\sqrt{\frac{2\pi v}{\alpha}}(\alpha T + s_0)\right) &\rightarrow \frac{1}{2} & C\left(\sqrt{\frac{2\pi v}{\alpha}}s_0\right) &\rightarrow 0 \\
S\left(\sqrt{\frac{2\pi v}{\alpha}}(\alpha T + s_0)\right) &\rightarrow \frac{1}{2} & S\left(\sqrt{\frac{2\pi v}{\alpha}}s_0\right) &\rightarrow 0 \\
\cos \frac{2\pi v s_0^2}{\alpha} &\rightarrow 1 & \sin \frac{2\pi v s_0^2}{\alpha} &\rightarrow 0
\end{aligned}$$

则式(6-17)变为:

$$H(u, v) \approx \frac{1}{2\sqrt{v\alpha}} \left[\frac{1}{2} - j \frac{1}{2} \right] = \frac{1-j}{4\sqrt{v\alpha}}$$

例6.7 如何推导天文图像退化过程的点扩散函数?

定义点扩散函数为当输入是点源时的图像系统的输出。在一幅天文图像中, 可以认为一颗非常遥远的星星就是一个点源。通过度量一颗星的亮度值, 可以迅速得到被研究图像退化过程的点扩散函数。

204

例6.8 假设在场景中有一条平行于图像轴 x 的理想亮度直线。利用这个信息来推导出使所获取图像退化的点扩散函数。

一条亮度直线的非退化图像在数学上可以表示为:

$$f(x, y) = \delta(y)$$

其中我们假设直线实际上是与 x 轴一致的。则这条直线的图像将变为:

$$h_l(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h(x - x', y - y') \delta(y') dy' dx' = \int_{-\infty}^{+\infty} h(x - x', y) dx'$$

变量代换 $\tilde{x} \equiv x - x' \Rightarrow dx' = -d\tilde{x}$ 。 \tilde{x} 的极限由 $+\infty$ 到 $-\infty$ 。那么:

$$h_l(x, y) = - \int_{+\infty}^{-\infty} h(\tilde{x}, y) d\tilde{x} = \int_{-\infty}^{+\infty} h(\tilde{x}, y) d\tilde{x} \quad (6-18)$$

等式右端并不依赖于 x , 因此等式左端也与 x 无关。即: 这条直线的图像会与 x 轴平行 (或者与 x 轴完全重合), 也就有:

$$h_l(x, y) = h_l(y) = \underbrace{\int_{-\infty}^{+\infty} h_l(\tilde{x}, y) d\tilde{x}}_{\tilde{x} \text{ 是独立于 } x \text{ 的虚拟变量}} \quad (6-19)$$

假设对 $h_l(y)$ 做傅里叶变换:

$$\hat{H}_l(v) \equiv \int_{-\infty}^{+\infty} h_l(y) e^{-2\pi j v y} dy \quad (6-20)$$

傅里叶变换后, 点扩散函数的传递函数由下式给出:

$$\hat{H}(u, v) = \int_{-\infty}^{+\infty} h(x, y) e^{-2\pi j (ux + vy)} dx dy \quad (6-21)$$

205

如果在该表达式中设 $u = 0$, 可得:

$$\hat{H}(0, v) = \int_{-\infty}^{+\infty} \underbrace{\left[\int_{-\infty}^{+\infty} h(x, y) dx \right]}_{h_l(y) \text{ 由式(6-19)给出}} e^{-2\pi j v y} dy \quad (6-22)$$

比较式(6-20)和式(6-22), 可得:

$$\hat{H}(0, v) = \hat{H}_l(v) \quad (6-23)$$

即, 理想直线的图像给出了传递函数沿单一方向的形状, 即垂直于直线的方向。这是容易理解的, 因为垂直于长度方向的线的截面不同于点的截面。按照定义, 一个点的截面就是指模糊过程的点扩散函数。如果图像中有足够多沿不同方向的理想直线, 就可获得传递函数在频域中沿垂直于直线方向的信息, 从而通过插值就可以计算频域中任意点的 $\hat{H}(u, v)$ 值。

例6.9 已知特定场景中有一条尖锐边缘。如何通过边缘图像来获得成像设备点扩散函数的信息呢?

假设理想边缘可以用沿 x 轴的阶跃函数来表示, 定义如下:

$$u(y) = \begin{cases} 1 & y > 0 \\ 0 & y \leq 0 \end{cases}$$

则函数图像为:

$$h_e(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x - x', y - y') u(y') dx' dy'$$

定义新变量 $\tilde{x} \equiv x - x'$, $\tilde{y} \equiv y - y'$ 。很明显, $dx' = -d\tilde{x}$, $dy' = -d\tilde{y}$ 。 \tilde{x} 和 \tilde{y} 的极限都是由 $+\infty$ 到 $-\infty$ 。那么:

$$h_e(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(\tilde{x}, \tilde{y}) u(y - \tilde{y}) d\tilde{x} d\tilde{y}$$

206

等式两边同时对 y 求偏导数:

$$\frac{\partial h_e(x, y)}{\partial y} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(\tilde{x}, \tilde{y}) \frac{\partial u(y - \tilde{y})}{\partial y} d\tilde{x} d\tilde{y}$$

已知对阶跃函数的变量求导后会得到delta函数:

$$\begin{aligned} \frac{\partial h_e(x, y)}{\partial y} &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(\tilde{x}, \tilde{y}) \delta(y - \tilde{y}) d\tilde{x} d\tilde{y} \Rightarrow \\ \frac{\partial h_e(x, y)}{\partial y} &= \int_{-\infty}^{\infty} h(\tilde{x}, y) d\tilde{x} \end{aligned} \quad (6-24)$$

比较式(6-24)和式(6-18), 可以看出对边缘图像求导后会得到一条线的图, 该线平行于原边缘。因此, 如果获得不同方向理想阶跃边缘的图像, 就能够推导出成像过程点扩散函数的信息。对每一幅阶跃边缘图像沿垂直于边缘的方向求微分。可以将每一幅得到的微分图看作是一条理想线的图像, 如例6.8所述, 它应该是点扩散函数沿该线的垂直方向所形成的线。

例6.10 使用例6.9的方法来推导成像设备的点扩散函数。

用直尺和黑墨水画一个测试图，如图6-4所示。

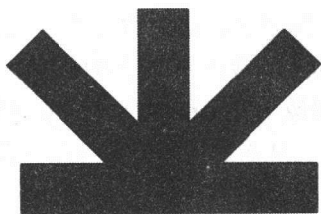


图6-4 推导成像设备点扩散函数的测试图

利用此测试图可以度量所使用的成像设备在 0° 、 45° 、 90° 和 135° 方向的点扩散函数。首先，用成像设备将测试图成像。然后使用Robinson算子与图像在上述四个方向进行卷积运算来求图像的偏导数，其中算子如图6-5所示。

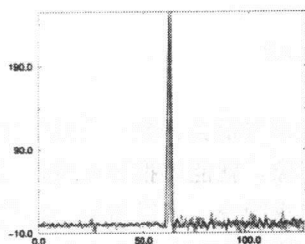
1	2	1	2	1	0	1	0	-1	0	-1	-2
0	0	0	1	0	-1	2	0	-2	1	0	-1
-1	-2	-1	0	-1	-2	1	0	-1	2	1	0

图6-5 用来计算 0° 、 45° 、 90° 和 135° 方向导数的滤波器

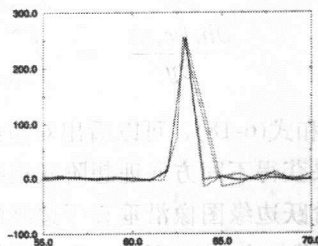
计算几条垂直于初始边缘线的合成图像，平均后就得到图6-6a所画的四条在 0° 、 45° 、 90° 和 135° 方向的曲线。这就是点扩散函数的曲线。而在图6-6b中，只是放大了图6-6a的中间部分。四条点扩散函数曲线中的两条明显窄于另外两条。这主要是由于它们对应于 45° 和 135° 方向，而在这两个方向上的像素距离是 0° 和 90° 方向上像素距离的 $\sqrt{2}$ 倍。所以，距离波峰1像素位置的点扩散函数值，实际上要显示在距离波峰将近1.4个像素的位置上。事实上，如果计算两对曲线的宽度比，就可以发现它的值是约等于1.4的。

图6-6c和图6-6d中分别给出了两对曲线，可以看出系统在 45° 、 135° 和 0° 、 90° 方向分别有着相同的表现形式。如果对 45° 和 135° 方向考虑上 $\sqrt{2}$ 的校正，就可以断定这个成像系统的点扩散函数是高度循环对称的。

在实际应用中，可以平均这四条曲线来产生一个循环对称点扩散函数的截面。这个二维函数的傅里叶变换就是成像设备的系统传递函数。



a) PSF的四条曲线



b) a)的放大图

图6-6 两种不同比例下成像系统的点扩散函数(PSF)

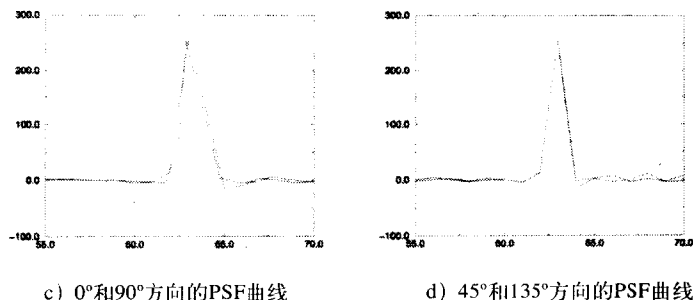


图6-6 (续)

•
•
•

6.14 如果知道退化过程的传递函数, 图像复原问题的解决方法是否就更简单

如果已知退化过程的传递函数, 然后计算退化图像的傅里叶变换, 那么由式(6-8)就可以得到非退化图像的傅里叶变换:

$$\hat{F}(u, v) = \frac{\hat{G}(u, v)}{\hat{H}(u, v)} \quad (6-25)$$

利用 $\hat{F}(u, v)$ 的傅里叶逆变换, 可以恢复 $f(x, y)$, 这正是我们想要的结果。但是这种直接的方法会产生令人难以接受的结果。

6.15 在 $\hat{H}(u, v) = 0$ 的点 (u, v) 会发生什么样的情况

在 (u, v) 平面的某些点处, $\hat{H}(u, v)$ 可能为0。那么, 从式(6-8)中可以得出在同一点的 $\hat{G}(u, v)$ 也将为0。这时式(6-25)中的 $\hat{G}(u, v) / \hat{H}(u, v)$ 将变为 $0/0$; 即不确定的。这意味着对特定频率 (u, v) , 原始图像的频率内容是恢复不了的。一种避免这种问题的简单方法就是忽略频域中的相应点, 只计算可以计算的点。

6.16 是否 $\hat{H}(u, v)$ 和 $\hat{G}(u, v)$ 的零点总是一致的

$\hat{H}(u, v)$ 和 $\hat{G}(u, v)$ 的零点并不总是一致的。如果式(6-8)中噪音数量非常微小, 那么 $\hat{H}(u, v)$ 的零点就与 $\hat{G}(u, v)$ 的零点不一致。

208
209

6.17 当我们写线性退化方程的时候如何把噪声考虑进去

对于加性噪声, 式(6-8)的完全形式为:

$$\hat{G}(u, v) = \hat{F}(u, v)\hat{H}(u, v) + \hat{N}(u, v) \quad (6-26)$$

其中, $\hat{N}(u, v)$ 项是噪声的傅里叶变换。于是 $\hat{F}(u, v)$ 可由下式给出:

$$\hat{F}(u, v) = \frac{\hat{G}(u, v)}{\hat{H}(u, v)} - \frac{\hat{N}(u, v)}{\hat{H}(u, v)} \quad (6-27)$$

当 $\hat{H}(u, v)$ 为0, 或者非常小时, 噪声项将被极度放大。

6.18 如何避免扩大噪声

许多情况下, $|\hat{H}(u, v)|$ 会从初始值开始快速递减, 而 $|\hat{N}(u, v)|$ 几乎总是常数。为了避免使用式(6-27)时引起噪声的扩大, 一般不直接将因子 $1/\hat{H}(u, v)$ 作为滤波器, 而将其加窗处理。在 $|\hat{H}(u, v)|$ 变得太小或者达到第一个零值前, 就将其在某一个频率截断。换句话说, 即:

$$\hat{F}(u, v) = \hat{M}(u, v)\hat{G}(u, v) - \hat{M}(u, v)\hat{N}(u, v) \quad (6-28)$$

其中

$$\hat{M}(u, v) = \begin{cases} 1/\hat{H}(u, v) & u^2 + v^2 \leq \omega_0^2 \\ 1 & u^2 + v^2 > \omega_0^2 \end{cases} \quad (6-29)$$

其中, 选择 ω_0 , 使得 $\hat{H}(u, v)$ 的所有零点都不包括在内。当然, 也可以用其他的窗口函数, 而

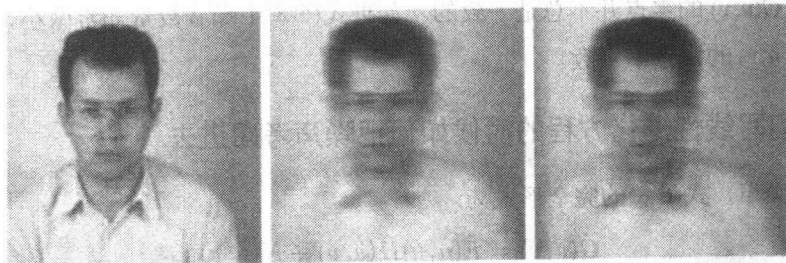
210 不采用上述矩形形状的窗口, 使 $\hat{M}(u, v)$ 在 ω_0 平稳地趋于0。

例6.11 通过复原运动模糊图像来说明逆滤波的实际应用。

考虑图6-7a。为了模仿图像在运动时模糊后的可能表现, 可以沿x轴方向, 每10个连续像素求一次平均值, 然后将其分配给第10个像素。这在下述情景中也是会发生的, 当摄像时, 相机往左边移动了10个像素的位置: 场景中一条长度等于10个像素的直线段的亮度可以由单个像素来记录。这个结果类似图6-7b。原始图像 $f(i, j)$ 模糊后得到的图像 $g(i, j)$ 由式(6-11)的离散形式给出:

$$g(i, j) = \frac{1}{i_T} \sum_{k=0}^{i_T-1} f(i-k, j) \quad i = 0, 1, \dots, N-1 \quad (6-30)$$

其中, i_T 为由同一个相机单元记录亮度的像素总数, 而 N 为图像一行中的像素数。在该例中, $i_T = 10$, $N = 128$ 。

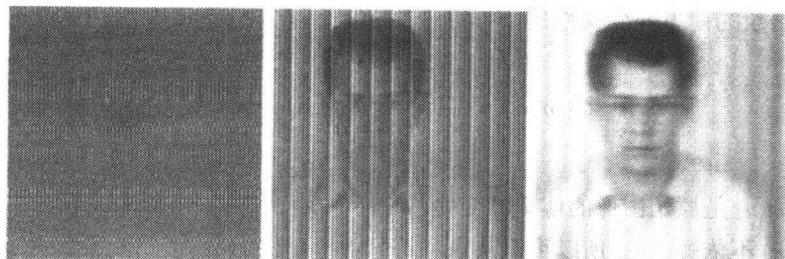


a) 原图

b) 实际模糊图像

c) 柱状边缘条件模糊图像

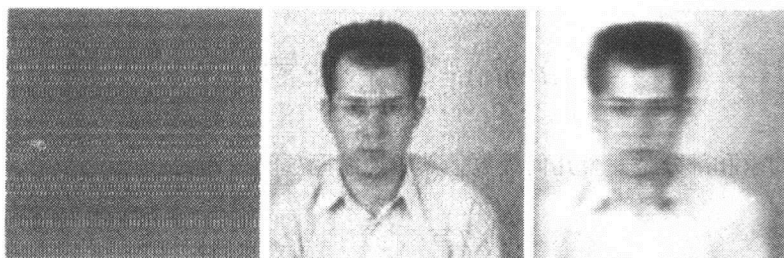
图6-7 用逆滤波进行图像复原



d) b) 的逆滤波

e) 忽略除0部分后
b) 的逆滤波

f) 忽略分母中第
一个0之外的部分
后 b) 的逆滤波



g) c) 的逆滤波

h) 忽略除0部分后c)
的逆滤波

i) 忽略分母中第一
个0之外的部分后c)
的逆滤波

图6-7 (续)

退化过程的传递函数可以由例6.4中推导的等式的离散形式来表示。下面来推导传递函数。 $g(i, j)$ 的离散傅里叶变换为:

$$\hat{G}(m, n) = \frac{1}{N} \sum_{l=0}^{N-1} \sum_{t=0}^{N-1} g(l, t) e^{-j(\frac{2\pi ml}{N} + \frac{2\pi nt}{N})} \quad (6-31)$$

将式(6-30)中的 $g(l, t)$ 代入, 可得:

$$\hat{G}(m, n) = \frac{1}{N} \frac{1}{i_T} \sum_{l=0}^{N-1} \sum_{t=0}^{N-1} \sum_{k=0}^{i_T-1} f(l-k, t) e^{-j(\frac{2\pi ml}{N} + \frac{2\pi nt}{N})}$$

重新安排求和顺序后, 等式变为:

$$\hat{G}(m, n) = \frac{1}{i_T} \sum_{k=0}^{i_T-1} \underbrace{\frac{1}{N} \sum_{l=0}^{N-1} \sum_{t=0}^{N-1} f(l-k, t) e^{-j(\frac{2\pi ml}{N} + \frac{2\pi nt}{N})}}_{\text{平移的 } f(l, t) \text{ 的 DFT}}$$

应用平移函数的傅里叶变换性质, 可得:

$$\hat{G}(m, n) = \frac{1}{i_T} \sum_{k=0}^{i_T-1} \hat{F}(m, n) e^{-j \frac{2\pi m}{N} k}$$

其中, $\hat{F}(m, n)$ 是原始图像的傅里叶变换。

由于 $\hat{F}(m, n)$ 不依赖于 k , 所以可以从求和符号中提出:

$$\hat{G}(m, n) = \hat{F}(m, n) \frac{1}{i_T} \sum_{k=0}^{i_T-1} e^{-j \frac{2\pi m}{N} k}$$

然后就可证明退化过程的傅里叶变换为,

$$\hat{H}(m, n) = \frac{1}{i_T} \sum_{k=0}^{i_T-1} e^{-j \frac{2\pi m}{N} k} \quad (6-32)$$

等式右边的求和部分是一个几何级数, 其中连续两项的比例为:

$$q \equiv e^{-j \frac{2\pi m}{N}}$$

应用公式:

$$\sum_{k=0}^{n-1} q^k = \frac{q^n - 1}{q - 1} \quad \text{其中 } q \neq 1$$

可得:

$$\hat{H}(m, n) = \frac{1}{i_T} \frac{e^{-j \frac{2\pi m}{N} i_T} - 1}{e^{-j \frac{2\pi m}{N}} - 1} = \frac{1}{i_T} \frac{e^{-j \frac{\pi m}{N} i_T} (e^{-j \frac{\pi m}{N} i_T} - e^{j \frac{\pi m}{N} i_T})}{e^{-j \frac{\pi m}{N}} (e^{-j \frac{\pi m}{N}} - e^{j \frac{\pi m}{N}})}$$

因此:

$$\hat{H}(m, n) = \frac{1}{i_T} \frac{\sin \frac{\pi m}{N} i_T}{\sin \frac{\pi m}{N}} e^{-j \frac{\pi m}{N} (i_T - 1)} \quad (6-33)$$

注意到当 $m = 0$ 时, 有 $q = 1$, 这样就不能应用几何级数的公式了。但可以利用式(6-32)中1的求和, 它等于 i_T , 则有:

$$\hat{H}(0, n) = 1, \text{ 其中 } 0 \leq n \leq N-1$$

比较式(6-33)和它对应的连续表达式(6-16), 可以看到两式间有一个根本区别: 在式(6-16)的分母中频率 u 沿模糊轴直接出现, 而在式(6-33)的分母中出现的是这个频率的正弦形式。这是由于离散图像被当作周期信号来做离散傅里叶变换, 因此, 在所有方向重复以至无穷。

我们可以分析模糊图像傅里叶变换后的实部和虚部:

$$\hat{G}(m, n) \equiv G_1(m, n) + jG_2(m, n)$$

其幅值幅角形式为:

$$\hat{G}(m, n) = \sqrt{G_1^2(m, n) + G_2^2(m, n)} e^{j\phi(m, n)} \quad (6-34)$$

其中

$$\begin{aligned} \cos \phi(m, n) &= \frac{G_1(m, n)}{\sqrt{G_1^2(m, n) + G_2^2(m, n)}} \\ \sin \phi(m, n) &= \frac{G_2(m, n)}{\sqrt{G_1^2(m, n) + G_2^2(m, n)}} \end{aligned} \quad (6-35)$$

为了得到原图的傅里叶变换, 将 $\hat{G}(m, n)$ 除以 $\hat{H}(m, n)$:

$$\hat{F}(m, n) = \frac{\sqrt{G_1^2(m, n) + G_2^2(m, n)}}{\sin \frac{i_T \pi m}{N}} i_T \sin \frac{\pi m}{N} e^{j(\phi(m, n) + \frac{\pi m}{N}(i_T - 1))}$$

因此, $\hat{F}(m, n)$ 的实部和虚部分别是 $F_1(m, n)$ 和 $F_2(m, n)$, 由下式给出:

$$\begin{aligned} F_1(m, n) &= i_T \sin \frac{\pi m}{N} \frac{\sqrt{G_1^2(m, n) + G_2^2(m, n)}}{\sin \frac{i_T \pi m}{N}} \cos \left(\phi(m, n) + \frac{\pi m}{N}(i_T - 1) \right) \\ F_2(m, n) &= i_T \sin \frac{\pi m}{N} \frac{\sqrt{G_1^2(m, n) + G_2^2(m, n)}}{\sin \frac{i_T \pi m}{N}} \sin \left(\phi(m, n) + \frac{\pi m}{N}(i_T - 1) \right) \end{aligned}$$

如果例用公式 $\cos(a+b) = \cos a \cos b - \sin a \sin b$ 和 $\sin(a+b) = \cos a \sin b + \sin a \cos b$, 并且将式(6-35)中 $\cos \phi(m, n)$ 和 $\sin \phi(m, n)$ 代入, 可得:

$$\begin{aligned} F_1(m, n) &= i_T \sin \frac{\pi m}{N} \frac{G_1(m, n) \cos \frac{\pi m(i_T - 1)}{N} - G_2(m, n) \sin \frac{\pi m(i_T - 1)}{N}}{\sin \frac{i_T \pi m}{N}} \\ F_2(m, n) &= i_T \sin \frac{\pi m}{N} \frac{G_1(m, n) \sin \frac{\pi m(i_T - 1)}{N} + G_2(m, n) \cos \frac{\pi m(i_T - 1)}{N}}{\sin \frac{i_T \pi m}{N}} \end{aligned} \quad (6-36)$$

可设:

$$F_1(0, n) = G_1(0, n) \quad 0 \leq n \leq N-1$$

$$F_2(0, n) = G_2(0, n) \quad 0 \leq n \leq N-1$$

如果将 $F_1(m, n)$ 和 $F_2(m, n)$ 作为非退化图像傅里叶变换的实部、虚部, 再进行傅里叶反变换, 就得到图像 6-7d。因为在式(6-36)中的一些 m 值是用 0 除的, 所以得到的图像最终是错误的。

事实上, 每次 $\frac{i_T \pi m}{N}$ 是 π 的倍数时, 分母 $\sin \frac{i_T \pi m}{N}$ 变成 0:

$$\frac{i_T \pi m}{N} = k\pi \Rightarrow m = \frac{kN}{i_T} \quad \text{其中} \quad k = 1, 2, \dots$$

图像大小是 128×128 , 即: $N = 128$, $i_T = 10$ 。因此, 当 $m = 12.8$ 、 25.6 、 38.4 等值的时候, 除的就是0。当 m 只能取整数值, 例如 13 、 26 、 38 等值的时候, 分母会变得非常小。只有当 $m = 64$ 时, 分母才会完全为0。可以忽略这个 m 值, 即, 设:

$$\begin{aligned} F_1(64, n) &= G_1(64, n) & 0 \leq n < 127 \\ F_2(64, n) &= G_2(64, n) & 0 \leq n < 127 \end{aligned}$$

其他的 $F_1(m, n)$ 和 $F_2(m, n)$ 的值是由式(6-36)来定义的。

如果进行傅里叶反变换, 就可以得到图6-7e中的图像。这时除了一些水平干涉频率外, 整个图像看上去几乎是可接受的。实际上, 我们并不是要确定使式(6-36)分母完全为0的所有 m 或 n 的值, 而是要找到这些值中的第一个值, 然后在应用公式时使 m 和 n 不大于这对值。本例

中, 在第一个零点处 $k = 1$, 即, $m = \frac{N}{i_T} = 12.8$ 。

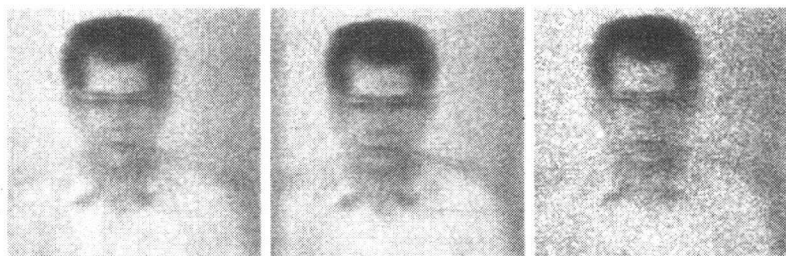
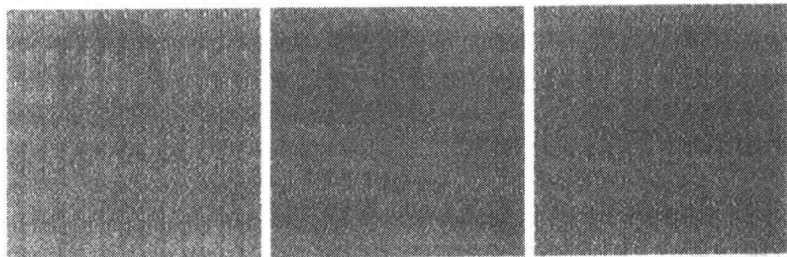
因此, 只有在 $0 \leq m < 12$ 和 $0 \leq n < 127$ 时, 才使用式(6-36)。否则, 就有:

$$\left. \begin{aligned} F_1(m, n) &= G_1(m, n) \\ F_2(m, n) &= G_2(m, n) \end{aligned} \right\} \quad \begin{aligned} 13 \leq m < 127 \\ 0 \leq n < 127 \end{aligned}$$

进行傅里叶反变换后, 得到的图像如图6-7f所示。这幅图看上去比前一幅有竖直线(水平干涉频率)的图更加模糊, 但水平干涉频率没有那么明显。产生模糊是可以理解的: 因为没有采取任何有效措施来增强 $m > 12$ 的频率部分, 所以对应于尖锐边缘的图像高频部分仍会保持退化。观察那些竖直线, 可以看出, 在一幅宽度为128的图像中大约有13条竖直线, 即每10个像素出现一条竖直线。这是由于边缘效应: 傅里叶变换假设图像在所有方向都是重复直至无穷的。所以它假设模糊图像左端的像素值等于图像右端的像素的真实值。当然, 实际上这是不可能的, 因为左端模糊像素所取的真实值是由一些更左端的点提供的, 而这些点在图像中是不出现的。为了证明这个解释是正确的, 我们假设柱状边缘条件将原始图像模糊, 即假设图像左端总是重复的。结果可得图6-7c的模糊图像。这幅图像以三种逆滤波方式复原后的结果可参见图6-7最下面的一行。竖直线已经完全消失了, 在图6-7h中已经获得了一个非常好的复原结果, 而这仅仅只是忽略了传递函数完全为0的频率。

但不幸的是, 在实际情况中出现的模糊, 总类似于图6-7b。而所得到的复原结果更接近于图6-7e和图6-7f而非图6-7h和图6-7i。

为了比较逆滤波器如何处理噪声, 采用图6-8中加入白高斯噪声的模糊加噪图像。然后使用逆滤波器并避免除0的项, 来复原噪声图像。图6-8d到图6-8f所示的结果非常不好: 由噪声控制的高频部分被滤波器放大, 使得复原图像中绝大部分为高频噪声信息。当滤波器超出第一个取0值的部分被截断后, 复原结果还是相当理想的, 如图6-8g到图6-8i所示。

a) 具有附加高斯噪声的实际模糊 ($\sigma=10$)b) 使用具有附加高斯噪声的柱状边缘条件后的模糊 ($\sigma=10$)c) 附加高斯噪声的实际模糊 ($\sigma=20$)

d) 忽略除0部分后a)的逆滤波

e) 忽略除0部分后b)的逆滤波

f) 忽略除0部分后c)的逆滤波



g) 忽略分母中第一个0之后a)的逆滤波

h) 忽略分母中第一个0之后b)的逆滤波

i) 忽略分母中第一个0之后c)的逆滤波

图6-8 噪声存在时用逆滤波进行图像复原

6.19 如何形式化地表达图像复原问题

如果 $\hat{f}(\mathbf{r})$ 是原未退化图像 $f(\mathbf{r})$ 的一幅估计图像, 我们希望通过计算 $\hat{f}(\mathbf{r})$, 使得在图像 $f(\mathbf{r})$ 的所有形式中, 残差图像 $f(\mathbf{r}) - \hat{f}(\mathbf{r})$ 的范数最小。由第3章可知, 这等价于确定 $\hat{f}(\mathbf{r})$ 使得下式最小化

$$e^2 \equiv E\{[f(\mathbf{r}) - \hat{f}(\mathbf{r})]^2\} \quad (6-37)$$

本节中, 假设式(6-37)有如下形式:

$$g(\mathbf{r}) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h(\mathbf{r} - \mathbf{r}') f(\mathbf{r}') d\mathbf{r}' + \nu(\mathbf{r}) \quad (6-38)$$

其中, $g(\mathbf{r})$ 、 $f(\mathbf{r})$ 和 $\nu(\mathbf{r})$ 都是随机场, 且 $\nu(\mathbf{r})$ 是噪声场。

6.20 式(6-37)的解是什么

如果等式的解没有附加任何条件, 那么可以最小化式(6-37)的 $f(\mathbf{r})$ 的最小平方估计, 将是给定 $g(\mathbf{r})$ 时 $f(\mathbf{r})$ 的条件期望 (conditional expectation)。通常它是 $g(\mathbf{r})$ 的非线性函数, 且需要计算随机场 $f(\mathbf{r})$ 和 $g(\mathbf{r})$ 的联合概率密度函数。这可以用类似模拟退火 (simulated annealing) 的非线性方法来计算。不过这类方法已超出了本书的研究范围。

6.21 可以求出式(6-37)的线性解吗

可以。只要加入约束条件: $\hat{f}(\mathbf{r})$ 的解是 $g(\mathbf{r})$ 的线性函数即可。很明显, 通过这种方式找到的解并不能使 e 绝对最小, 但可以使 e 在约束限制的条件下达到最小。希望将估计图像 $\hat{f}(\mathbf{r})$ 表示为退化图像灰度级的线性函数, 即:

$$\hat{f}(\mathbf{r}) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} m(\mathbf{r}, \mathbf{r}') g(\mathbf{r}') d\mathbf{r}' \quad (6-39)$$

其中, $m(\mathbf{r}, \mathbf{r}')$ 是需要确定的函数, 它给出了退化图像 g 在 \mathbf{r}' 处的灰度级值对估计图像 \hat{f} 在 \mathbf{r} 处的值影响的权值。如果随机场是齐次的, 则该权值函数 $m(\mathbf{r}, \mathbf{r}')$ 只依赖于 \mathbf{r} 和 \mathbf{r}' 的差, 而不是单独依赖于各自的值。所以式(6-39)可写为:

$$\hat{f}(\mathbf{r}) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} m(\mathbf{r} - \mathbf{r}') g(\mathbf{r}') d\mathbf{r}' \quad (6-40)$$

上式意味着为了获得非退化图像 $f(\mathbf{r})$ 的估计图像 $\hat{f}(\mathbf{r})$, 需要确定与退化图像 $g(\mathbf{r})$ 卷积的滤波器 $m(\mathbf{r})$ 。

6.22 图像复原问题的线性最小均方误差解是什么

设 $\hat{M}(u, v)$ 是滤波器 $m(\mathbf{r})$ 的傅里叶变换, 如果其满足式(6-41), 则可求得式(6-37)的线性解。

$$\hat{M}(u, v) = \frac{S_{fg}(u, v)}{S_{gg}(u, v)} \quad (6-41)$$

其中 $S_{fg}(u, v)$ 为非退化图像和退化图像的互谱密度函数, $S_{gg}(u, v)$ 是退化图像的谱密度函数。 $\hat{M}(u, v)$ 是Wiener滤波器的傅里叶变换, 用来实现图像复原。

6.23 如果原图像 $f(\mathbf{r})$ 未知, 如何利用依赖于退化图像的互谱密度函数的式(6-41)来推导需要的滤波器

为了便于处理, 需要做一些额外假设: 噪声和实际图像是不相关的, 并且两者中至少有一个有零均值。这个假设似乎是合理的: 我们希望增强图像的过程与增大噪声的过程是完全不相同的。此外, 如果噪声有偏差, 即噪声均值非零, 通常我们要确定并且减去这个偏差使得它有零均值。

因为 $f(\mathbf{r})$ 和 $\nu(\mathbf{r})$ 不相关, 且 $E\{\nu(\mathbf{r})\} = 0$, 则:

$$E\{f(\mathbf{r})\nu(\mathbf{r})\} = E\{f(\mathbf{r})\}E\{\nu(\mathbf{r})\} = 0 \quad (6-42)$$

为了得到原图像和退化图像的互谱密度函数, 我们将式(6-38)两边同乘以 $f(\mathbf{r}-\mathbf{s})$ 再取期望值:

$$\underbrace{E\{g(\mathbf{r})f(\mathbf{r}-\mathbf{s})\}}_{R_{gf}(\mathbf{s})} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h(\mathbf{r}-\mathbf{r}') \underbrace{E\{f(\mathbf{r}')f(\mathbf{r}-\mathbf{s})\}}_{R_{ff}(\mathbf{r}'-\mathbf{r}+\mathbf{s})} d\mathbf{r}' + \underbrace{E\{f(\mathbf{r}-\mathbf{s})\nu(\mathbf{r})\}}_{\text{由式(6-42)知该项为0}}$$

因此:

$$R_{gf}(\mathbf{s}) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h(\mathbf{r}-\mathbf{r}') R_{ff}(\mathbf{r}'-\mathbf{r}+\mathbf{s}) d\mathbf{r}'$$

上式的傅里叶变换为 (参见方框B6.4和例6.12):

$$S_{gf}(u, v) = \hat{H}^*(u, v) S_{ff}(u, v) \quad (6-43)$$

从式(6-38)中, 可以看出 (参见方框B6.3)

$$S_{gg}(u, v) = S_{ff}(u, v) |\hat{H}(u, v)|^2 + S_{\nu\nu}(u, v) \quad (6-44)$$

如果将式(6-43)和式(6-44)代入式(6-41), 可得:

$$\hat{M}(u, v) = \frac{\hat{H}^*(u, v) S_{ff}(u, v)}{S_{ff}(u, v) |\hat{H}(u, v)|^2 + S_{\nu\nu}(u, v)} \quad (6-45)$$

或者

$$\hat{M}(u, v) = \frac{\hat{H}^*(u, v)}{|\hat{H}(u, v)|^2 + \frac{S_{\nu\nu}(u, v)}{S_{ff}(u, v)}} \quad (6-46)$$

如果分子、分母同乘以 $\hat{H}(u, v)$, 可得:

$$\hat{M}(u, v) = \frac{1}{\hat{H}(u, v)} \frac{|\hat{H}(u, v)|^2}{|\hat{H}(u, v)|^2 + \frac{S_{\nu\nu}(u, v)}{S_{ff}(u, v)}} \quad (6-47)$$

上式给出了Wiener滤波器的傅里叶变换, 可以用来进行图像复原。

6.24 如果我们对于未知图像 $f(\mathbf{r})$ 的统计信息完全不知道, 应该如何使用式(6-47)

如果完全不知道图像的统计信息, 但仍想要复原, 即, 不知道 $S_{ff}(u, v)$, 则可以用常数 Γ 来代替式(6-47)中的 $\frac{S_{vv}(u, v)}{S_{ff}(u, v)}$, 再对不同的 Γ 值进行实验。

因为 $\frac{S_{vv}(u, v)}{S_{ff}(u, v)}$ 是 (u, v) 的函数而不是常数, 所以这样做有点过度简单化。

6.25 Wiener滤波器式(6-47)和式(6-25)的逆滤波器之间的关系是怎样的

当噪声不存在时, $S_{vv}(u, v) = 0$, Wiener滤波器变为式(6-25)的逆传递函数滤波器。线性最小平方误差方法只是给出了一个修正因子, 考虑噪声的影响时, 退化过程的逆传递函数要乘上这个因子才可以作为滤波器。

6.26 假定知道未知图像 $f(\mathbf{r})$ 的统计信息, 能否确定由 $S_w(\mathbf{r})$ 所表达的噪声的统计信息

通常假设噪声为白噪声; 即

$$S_{vv}(u, v) = \text{常数} = S_{vv}(0, 0) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} R_{vv}(x, y) dx dy \quad (6-48)$$

如果再设噪声是遍历的, 就可以用一张纯噪声图像计算得到 $R_{vv}(x, y)$, 其中纯噪声图像 $g(x, y)$ 是在没有原图像即 $f(x, y) = 0$ 时记录的。

B6.1 如果 $m(\mathbf{r}-\mathbf{r}')$ 满足

$$E \left\{ \left[f(\mathbf{r}) - \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} m(\mathbf{r}-\mathbf{r}') g(\mathbf{r}') d\mathbf{r}' \right] g(\mathbf{s}) \right\} = 0 \quad (6-49)$$

则可使式(6-37)中定义的误差达到最小值。

如果将式(6-40)代入式(6-37), 可得:

$$e^2 = E \left\{ \left[f(\mathbf{r}) - \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} m(\mathbf{r}-\mathbf{r}') g(\mathbf{r}') d\mathbf{r}' \right]^2 \right\} \quad (6-50)$$

再来考虑另一个不满足式(6-49)的函数 $m'(\mathbf{r})$ 。用它来进行图像复原时, 将生成一幅误差为 e'^2 的估计图像, 这个误差比由 $m(\mathbf{r})$ 所得的估计图像的误差要大。其中 e'^2 满足:

$$e'^2 \equiv E \left\{ \left[f(\mathbf{r}) - \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} m'(\mathbf{r}-\mathbf{r}') g(\mathbf{r}') d\mathbf{r}' \right]^2 \right\}$$

在被积函数中, 对函数 $m'(\mathbf{r}-\mathbf{r}')$ 先加上再减去函数 $m(\mathbf{r}-\mathbf{r}')$, 就可以将积分分为两部分, 然后再展开平方项:

$$\begin{aligned}
e'^2 &= E \left\{ \left[f(\mathbf{r}) - \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} [m'(\mathbf{r}-\mathbf{r}') + m(\mathbf{r}-\mathbf{r}') - m(\mathbf{r}-\mathbf{r}')] g(\mathbf{r}') d\mathbf{r}' \right]^2 \right\} \\
&= E \left\{ \left[\left(f(\mathbf{r}) - \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} m(\mathbf{r}-\mathbf{r}') g(\mathbf{r}') d\mathbf{r}' \right) \right. \right. \\
&\quad \left. \left. + \left(\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} [m(\mathbf{r}-\mathbf{r}') - m'(\mathbf{r}-\mathbf{r}')] g(\mathbf{r}') d\mathbf{r}' \right) \right]^2 \right\} \\
&= E \left\{ \underbrace{\left(f(\mathbf{r}) - \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} m(\mathbf{r}-\mathbf{r}') g(\mathbf{r}') d\mathbf{r}' \right)^2}_{e^2} \right\} \\
&\quad + E \left\{ \underbrace{\left(\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} [m(\mathbf{r}-\mathbf{r}') - m'(\mathbf{r}-\mathbf{r}')] g(\mathbf{r}') d\mathbf{r}' \right)^2}_{\text{非负数}} \right\} \\
&\quad + 2E \left\{ \underbrace{\left(f(\mathbf{r}) - \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} m(\mathbf{r}-\mathbf{r}') g(\mathbf{r}') d\mathbf{r}' \right) \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} [m(\mathbf{r}-\mathbf{r}') - m'(\mathbf{r}-\mathbf{r}')] g(\mathbf{r}') d\mathbf{r}'}_{\text{令}} \right\}
\end{aligned} \tag{6-51}$$

第一项的期望值是 e^2 ，第二项的期望值一定是非负数。将最后一项第二个因子中的积分变量 \mathbf{r}' 转换为 s 。则式(6-51)右端最后一项就可写为：

$$2E \left\{ \left(f(\mathbf{r}) - \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} m(\mathbf{r}-\mathbf{r}') g(\mathbf{r}') d\mathbf{r}' \right) \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} [m(\mathbf{r}-\mathbf{s}) - m'(\mathbf{r}-\mathbf{s})] g(\mathbf{s}) d\mathbf{s} \right\}$$

上式中的第一个因子不依赖于 s ，因此可以放入 \iint 内：

$$2E \left\{ \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \left(f(\mathbf{r}) - \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} m(\mathbf{r}-\mathbf{r}') g(\mathbf{r}') d\mathbf{r}' \right) [m(\mathbf{r}-\mathbf{s}) - m'(\mathbf{r}-\mathbf{s})] g(\mathbf{s}) d\mathbf{s} \right\}$$

$[m(\mathbf{r}-\mathbf{s}) - m'(\mathbf{r}-\mathbf{s})]$ 的差只是两个具体函数的差，而不是随机场。如果改变积分和取期望值的顺序，这个期望值不会影响该因子，所以这项可以写为：

$$2 \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} E \left\{ \left[f(\mathbf{r}) - \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} m(\mathbf{r}-\mathbf{r}') g(\mathbf{r}') d\mathbf{r}' \right] g(\mathbf{s}) \right\} [m(\mathbf{r}-\mathbf{s}) - m'(\mathbf{r}-\mathbf{s})] d\mathbf{s}$$

但是根据式(6-49)，上式中的期望值为0，所以式(6-51)可写为：

$$e'^2 = e^2 + \text{非负项}$$

由此可知，函数 $m'(\mathbf{r})$ 产生的复原误差总是大于等于 $m(\mathbf{r})$ 产生的复原误差。所以，满足式(6-49)的 $m(\mathbf{r})$ 可以使式(6-37)中定义的误差达到最小。

• 例6.12(B) 设 $\hat{F}(u, v)$ 、 $\hat{H}(u, v)$ 和 $\hat{G}(u, v)$ 分别为 $f(\mathbf{r})$ 、 $h(\mathbf{r})$ 和 $g(\mathbf{r})$ 的傅里叶变换, 且

$$g(\mathbf{r}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(\mathbf{t} - \mathbf{r}) f(\mathbf{t}) d\mathbf{t} \quad (6-52)$$

则可证明

$$\hat{G}(u, v) = \hat{H}^*(u, v) \hat{F}(u, v) \quad (6-53)$$

其中 $\hat{H}^*(u, v)$ 是 $\hat{H}(u, v)$ 的复共轭 (complex conjugate)。

假设 $\mathbf{r} = (x, y)$, $\mathbf{t} = (\tilde{x}, \tilde{y})$ 。三个函数的傅里叶变换为:

$$\hat{G}(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) e^{-j(ux+vy)} dx dy \quad (6-54)$$

$$\hat{F}(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j(ux+vy)} dx dy \quad (6-55)$$

$$\hat{H}(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x, y) e^{-j(ux+vy)} dx dy \quad (6-56)$$

$\hat{H}(u, v)$ 的复共轭为

$$\hat{H}^*(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x, y) e^{j(ux+vy)} dx dy \quad (6-57)$$

将式(6-52)中的 $g(x, y)$ 代入式(6-54)的右端:

$$\hat{G}(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(\tilde{x} - x, \tilde{y} - y) f(\tilde{x}, \tilde{y}) d\tilde{x} d\tilde{y} e^{-j(ux+vy)} dx dy$$

定义积分新变量 $s_1 \equiv \tilde{x} - x$, $s_2 \equiv \tilde{y} - y$ 代替原来的 x 和 y 。因为 $dx = -ds_1$, $dy = -ds_2$, 则 $dx dy = ds_1 ds_2$ 。又因为 s_1 和 s_2 的极限都是从 $+\infty$ 到 $-\infty$, 所以, 我们可以交换它们的顺序而不用考虑符号的改变

$$\hat{G}(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(s_1, s_2) f(\tilde{x}, \tilde{y}) d\tilde{x} d\tilde{y} e^{-j(u(\tilde{x}-s_1)+v(\tilde{y}-s_2))} ds_1 ds_2$$

两个双积分是可分离的:

$$\hat{G}(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(s_1, s_2) e^{j(us_1+vs_2)} ds_1 ds_2 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\tilde{x}, \tilde{y}) e^{-j(u\tilde{x}+v\tilde{y})} d\tilde{x} d\tilde{y}$$

可以看出, 等式右端的结果分别是式(6-55)中的 $\hat{F}(u, v)$ 和式(6-57)中的 $\hat{H}^*(u, v)$ 。从而证明了式(6-53)。

B6.2 证明随机场 $f(x, y)$ 的空间自相关函数的傅里叶变换等于随机场的频谱密度 $|\hat{F}(u, v)|^2$ (Wiener-Khinchine 定理)。

定义 $f(x, y)$ 的空间自相关函数为

$$R_{ff}(\tilde{x}, \tilde{y}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x + \tilde{x}, y + \tilde{y}) f(x, y) dx dy \quad (6-58)$$

将式(6-58)的两边同时乘以傅里叶变换核, 求积分, 即可得到 $R_{ff}(\tilde{x}, \tilde{y})$ 的傅里叶变换 $\hat{R}_{ff}(u, v)$:

$$\begin{aligned} \hat{R}_{ff}(u, v) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} R_{ff}(\tilde{x}, \tilde{y}) e^{-j(\tilde{x}u + \tilde{y}v)} d\tilde{x} d\tilde{y} \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x + \tilde{x}, y + \tilde{y}) f(x, y) e^{-j(\tilde{x}u + \tilde{y}v)} dx dy d\tilde{x} d\tilde{y} \end{aligned} \quad (6-59)$$

223

定义新的积分变量 $s_1 \equiv x + \tilde{x}$ 和 $s_2 \equiv y + \tilde{y}$, 代替原来的积分变量 \tilde{x} 和 \tilde{y} 。那么 $\tilde{x} = s_1 - x$, $\tilde{y} = s_2 - y$, $d\tilde{x}d\tilde{y} = ds_1 ds_2$, 并且积分极限也没有改变:

$$\hat{R}_{ff}(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(s_1, s_2) f(x, y) e^{-j((s_1-x)u + (s_2-y)v)} dx dy ds_1 ds_2$$

右端的两个双积分是可分离的, 所以可以写为:

$$\hat{R}_{ff}(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(s_1, s_2) e^{-j(s_1 u + s_2 v)} ds_1 ds_2 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{j(xu + yv)} dx dy$$

可以看到, 此式右端的第一个双积分是 $f(s_1, s_2)$ 的傅里叶变换 $\hat{F}(u, v)$, 第二个双积分为其复共轭 $\hat{F}^*(u, v)$ 。所以有:

$$\hat{R}_{ff}(u, v) = \hat{F}(u, v) \hat{F}^*(u, v) = |\hat{F}(u, v)|^2$$

B6.3 推导能使式(6-37)中定义的误差达到最小的函数 $m(\mathbf{r})$ 的傅里叶变换表达式。

最小化式(6-37)的 $m(\mathbf{r})$ 满足式(6-49)时, 该式可写为:

$$E\{f(\mathbf{r})g(\mathbf{s})\} - E\left\{\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} m(\mathbf{r} - \mathbf{r}') g(\mathbf{r}') g(\mathbf{s}) d\mathbf{r}'\right\} = 0$$

其中, 因为 $g(\mathbf{s})$ 不依赖于 \mathbf{r}' , 所以放入到积分号内。该式第二项中的期望算子实际上只与随机函数 $g(\mathbf{r}')$ 和 $g(\mathbf{s})$ 有关。因此, 可以写为:

$$\underbrace{E\{f(\mathbf{r})g(\mathbf{s})\}}_{R_{fg}(\mathbf{r}, \mathbf{s})} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} m(\mathbf{r} - \mathbf{r}') \underbrace{E\{g(\mathbf{r}')g(\mathbf{s})\}}_{R_{gg}(\mathbf{r}', \mathbf{s})} d\mathbf{r}' \quad (6-60)$$

按照随机场的自相关和互相关函数的定义,有:

$$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} m(\mathbf{r} - \mathbf{r}') R_{gg}(\mathbf{r}', \mathbf{s}) d\mathbf{r}' = R_{fg}(\mathbf{r}, \mathbf{s}) \quad (6-61)$$

如前所知 (参见例3.7), 齐次随机场上的相关函数可以写成该函数两个变量的差的函数, 所以:

$$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} m(\mathbf{r} - \mathbf{r}') R_{gg}(\mathbf{r}' - \mathbf{s}) d\mathbf{r}' = R_{fg}(\mathbf{r} - \mathbf{s}) \quad (6-62)$$

加入新的变量: $\mathbf{r}' - \mathbf{s} = \mathbf{t}$ 和 $\mathbf{r} - \mathbf{s} = \boldsymbol{\tau}$. 于是 $d\mathbf{r}' = d\mathbf{t}$, $\mathbf{r} - \mathbf{r}' = \boldsymbol{\tau} - \mathbf{t}$. 那么:

$$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} m(\boldsymbol{\tau} - \mathbf{t}) R_{gg}(\mathbf{t}) d\mathbf{t} = R_{fg}(\boldsymbol{\tau}) \quad (6-63)$$

这是退化图像的自相关函数和所要求的滤波器之间的卷积。根据卷积定理, 该式等价于两个函数傅里叶变换的乘积:

$$\hat{M}(u, v) S_{gg}(u, v) = S_{fg}(u, v) \quad (6-64)$$

其中, S_{gg} 和 S_{fg} 分别是退化图像的谱密度函数和退化图像与非退化图像的互谱密度函数。即, 分别为函数 g 的自相关函数的傅里叶变换和函数 f 与函数 g 的互相关函数的傅里叶变换。因此:

$$\hat{M}(u, v) = \frac{S_{fg}(u, v)}{S_{gg}(u, v)} \quad (6-65)$$

能使实际图像与重建图像之间均方差最小的最佳复原滤波器, 其傅里叶变换等于退化图像与真实图像之间的互谱密度和退化图像的谱密度之间的比率。

B6.4 如果齐次随机场 $g(x, y)$, $f(x, y)$ 和 $\nu(x, y)$ 的谱密度函数分别为 $S_{gg}(u, v)$, $S_{ff}(u, v)$ 和 $S_{\nu\nu}(u, v)$, $\hat{H}(u, v)$ 是 $h(x, y)$ 的傅里叶变换, 且

$$g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x - \tilde{x}, y - \tilde{y}) f(\tilde{x}, \tilde{y}) d\tilde{x} d\tilde{y} + \nu(x, y) \quad (6-66)$$

证明当 $f(x, y)$ 与 $\nu(x, y)$ 不相关, 且两者至少有一个有零均值时, 有

$$S_{gg}(u, v) = S_{ff}(u, v) |H(u, v)|^2 + S_{\nu\nu}(u, v) \quad (6-67)$$

如果式(6-66)两边同乘以 $g(x+s_1, y+s_2)$, 且对随机场 $g(x, y)$ 的所有变量整体取均值, 则有:

$$\begin{aligned} E\{g(x, y)g(x+s_1, y+s_2)\} &= E\left\{g(x+s_1, y+s_2) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x-\tilde{x}, y-\tilde{y}) f(\tilde{x}, \tilde{y}) d\tilde{x} d\tilde{y}\right\} \\ &\quad + E\{g(x+s_1, y+s_2)\nu(x, y)\} \end{aligned}$$

因为 $g(x, y)$ 是一个齐次随机场, 可以看出, 等式的左端是位移变量为 s_1 和 s_2 的 g 的自相关函数 $R_{gg}(s_1, s_2)$ 。噪声随机场 $v(x, y)$ 也是齐次的, 所以右端最后一项是随机场 g 和 v 的互相关函数 $R_{gv}(s_1, s_2)$ 。此外, $g(x + s_1, y + s_2)$ 不依赖于积分变量 \tilde{x} 和 \tilde{y} , 所以可以放入右端第一项的积分号内:

$$R_{gg}(s_1, s_2) = E \left\{ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x - \tilde{x}, y - \tilde{y}) f(\tilde{x}, \tilde{y}) g(x + s_1, y + s_2) d\tilde{x} d\tilde{y} \right\} + R_{gv}(s_1, s_2)$$

求期望值和积分是两个可以相互交换的线性运算。期望算子只在随机场 f 和 g 上计算, 函数 h 并不受影响。因此该式可写为:

$$R_{gg}(s_1, s_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x - \tilde{x}, y - \tilde{y}) E \{ f(\tilde{x}, \tilde{y}) g(x + s_1, y + s_2) \} d\tilde{x} d\tilde{y} + R_{gv}(s_1, s_2)$$

可以看到, 积分号内实际是 f 和 g 的互相关函数 R_{gf} 在位移值为 $x + s_1 - \tilde{x}$ 和 $y + s_2 - \tilde{y}$ 时的值:

$$R_{gg}(s_1, s_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x - \tilde{x}, y - \tilde{y}) R_{gf}(x - \tilde{x} + s_1, y - \tilde{y} + s_2) d\tilde{x} d\tilde{y} + R_{gv}(s_1, s_2) \quad (6-68)$$

重新定义积分变量: $x - \tilde{x} = \alpha$, $y - \tilde{y} = \beta$ 。那么 $d\tilde{x}d\tilde{y} = d\alpha d\beta$, 且两个积分极限的符号变化互相抵消:

$$R_{gg}(s_1, s_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(\alpha, \beta) R_{gf}(\alpha + s_1, \beta + s_2) d\alpha d\beta + R_{gv}(s_1, s_2)$$

再改变积分变量为 $w = \alpha + s_1, z = \beta + s_2$ 。那么 $\alpha = w - s_1, \beta = z - s_2, d\alpha d\beta = dw dz$, 且积分极限不受影响:

$$R_{gg}(s_1, s_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(w - s_1, z - s_2) R_{gf}(w, z) dw dz + R_{gv}(s_1, s_2) \quad (6-69)$$

如果上式两边同取傅里叶变换, 利用例6.12的结果, 则有:

$$\hat{R}_{gg}(u, v) = \hat{H}^*(u, v) \hat{R}_{gf}(u, v) + \hat{R}_{gv}(u, v)$$

其中带有 $(\hat{})$ 的量表示式(6-69)中出现的相应量的傅里叶变换。

如果随机场是遍历的, 这里计算的整体自相关和互相关函数, 就应该与空间自相关和互相关函数是相同的。这些函数的傅里叶变换就是相应随机场的谱密度函数或者互谱密度函数 (参见方框B6.2)。然而, 一般情况下, 这些随机场通常都不是遍历的。那么就只能假设自相关和互相关函数的傅里叶变换分别是相应随机场的谱密度和互谱密度。

因此, 必须注意, 在设计Wiener滤波器时, 总是默认遍历性的假设。有了这个前提, 上式可以写为:

$$S_{gg}(u, v) = \hat{H}^*(u, v) S_{gf}(u, v) + S_{gv}(u, v) \quad (6-70)$$

其中, 需要计算随机场 f 和 g 间的互谱密度函数。再从式(6-66)开始, 但两边同乘以 $f(x-s_1, y-s_2)$, 然后取期望。之所以乘以 $f(x-s_1, y-s_2)$ 而不是 $f(x+s_1, y+s_2)$, 主要因为在推导式(6-68)中 R_{gf} 的位移变量时, 是用 g 的变量减去 f 的变量。所以这里再次遵循原有的惯例。处理过程和前面一样, 则有:

$$R_{gf}(s_1, s_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x-\tilde{x}, y-\tilde{y}) E\{f(\tilde{x}, \tilde{y})f(x-s_1, y-s_2)\} d\tilde{x}d\tilde{y} \\ + E\{f(x-s_1, y-s_2)\nu(x, y)\} \quad (6-71)$$

由于随机场 f 和 ν 是不相关的, 并且至少其中之一有0均值。那么:

$$E\{f(x-s_1, y-s_2)\nu(x, y)\} = E\{f(x-s_1, y-s_2)\} E\{\nu(x, y)\} = 0 \quad (6-72)$$

可以看出, 在式(6-71)右端的积分中, 包含随机场 f 在位移变量为 $(\tilde{x}-x+s_1, \tilde{y}-y+s_2)$ 的自相关函数。这里用 $f(\tilde{x}, \tilde{y})$ 的变量减去 $f(x-s_1, y-s_2)$ 的变量, 是因为在等式的左端, 我们是用已有函数的变量减去“新”函数的变量(即由 $g(x, y)$ 的变量减去 $f(x-s_1, y-s_2)$ 的变量)来构造 R_{gf} 的。所以有:

$$R_{gf}(s_1, s_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x-\tilde{x}, y-\tilde{y}) R_{ff}(\tilde{x}-x+s_1, \tilde{y}-y+s_2) d\tilde{x}d\tilde{y}$$

定义新的积分变量 $\alpha = x - \tilde{x}$, $\beta = y - \tilde{y}$:

$$R_{gf}(s_1, s_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(\alpha, \beta) R_{ff}(s_1 - \alpha, s_2 - \beta) d\alpha d\beta$$

上式是一个简单的卷积, 其傅里叶变换可以写为:

$$\hat{R}_{gf}(u, v) = \hat{H}(u, v) \hat{R}_{ff}(u, v)$$

再次利用相关函数的傅里叶变换与谱密度函数相等的假设, 可得:

$$\hat{S}_{gf}(u, v) = \hat{H}(u, v) \hat{S}_{ff}(u, v) \quad (6-73)$$

式(6-68)中也需要知道随机场 g 和 ν 的互谱密度函数。那么再从式(6-66)开始两边同乘以 $\nu(x-s_1, y-s_2)$, 取期望值:

$$\hat{R}_{g\nu}(s_1, s_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x-\tilde{x}, y-\tilde{y}) E\{f(\tilde{x}, \tilde{y})\nu(x-s_1, y-s_2)\} d\tilde{x}d\tilde{y} + R_{\nu\nu}(s_1, s_2)$$

由式(6-72)可知, 积分项的值为0。上式求傅里叶变换后为:

$$\hat{S}_{g\nu}(u, v) = \hat{S}_{\nu\nu}(u, v) \quad (6-74)$$

如果将式(6-73), 式(6-74)代入式(6-70), 即可得到要证明的式(6-67)。

• 例6.13 通过复原一幅运动模糊图像说明怎样应用Wiener滤波器。

• 我们再来考虑图6-7a中的模糊图像。由式(6-33)可得:

$$|\hat{H}(m, n)|^2 = \frac{1}{i_T^2 \sin^2 \frac{\pi m}{N}} \sin^2 \frac{i_T \pi m}{N} \quad [228]$$

对于式(6-46)给出的Wiener滤波, 我们用一个常数 Γ 来替换它的分母中频谱密度的比率, 则得到的Wiener滤波器为:

$$\hat{M}(m, n) = \frac{\frac{1}{i_T} \sin \frac{\pi m}{N} i_T e^{j \frac{\pi m (i_T - 1)}{N}}}{\frac{1}{i_T^2} \sin^2 \frac{\pi m}{N} i_T + \Gamma}$$

或者

$$\hat{M}(m, n) = \frac{i_T \sin \frac{\pi m}{N} \sin \frac{\pi m}{N} i_T e^{j \frac{\pi m (i_T - 1)}{N}}}{\sin^2 \frac{\pi m}{N} i_T + \Gamma i_T^2 \sin^2 \frac{\pi m}{N}}$$

对于 $m = 0$ 时的情况需格外注意, 这时有:

$$\hat{M}(0, n) = \frac{1}{1 + \Gamma}, \text{ 其中 } 0 \leq n \leq N-1$$

如果将该函数与由式(6-34)定义的模糊图像的傅里叶变换相乘, 可以得到:

$$\hat{F}(m, n) = \frac{i_T \sin \frac{\pi m}{N} \sin \frac{i_T \pi m}{N} \sqrt{G_1^2(m, n) + G_2^2(m, n)} e^{j(\phi(m, n) + \frac{(i_T - 1)\pi m}{N})}}{\sin^2 \frac{i_T \pi m}{N} + \Gamma i_T^2 \sin^2 \frac{\pi m}{N}}$$

对于 $m = 0$ 的情况, 我们有:

$$\hat{F}(0, n) = \frac{\sqrt{G_1^2(0, n) + G_2^2(0, n)}}{1 + \Gamma} e^{j\phi(0, n)} \text{ 其中 } 0 \leq n \leq N-1$$

$\hat{F}(m, n)$ 的实部和虚部分别为:

$$F_1(m, n) = \frac{i_T \sin \frac{\pi m}{N} \sin \frac{i_T \pi m}{N} \sqrt{G_1^2(m, n) + G_2^2(m, n)}}{\sin^2 \frac{i_T \pi m}{N} + \Gamma i_T^2 \sin^2 \frac{\pi m}{N}} \cos\left(\phi(m, n) + \frac{(i_T - 1)\pi m}{N}\right)$$

$$F_2(m, n) = \frac{i_T \sin \frac{\pi m}{N} \sin \frac{i_T \pi m}{N} \sqrt{G_1^2(m, n) + G_2^2(m, n)}}{\sin^2 \frac{i_T \pi m}{N} + \Gamma i_T^2 \sin^2 \frac{\pi m}{N}} \sin\left(\phi(m, n) + \frac{(i_T - 1)\pi m}{N}\right) \quad [229]$$

如果利用公式 $\cos(a + b) = \cos a \cos b - \sin a \sin b$ 和 $\sin(a + b) = \sin a \cos b + \cos a \sin b$, 并用式(6-35)定义的 $\cos \phi(m, n)$ 和 $\sin \phi(m, n)$ 代入, 可以得到:

$$F_1(m, n) = \frac{i_T \sin \frac{\pi m}{N} \sin \frac{i_T \pi m}{N}}{\sin^2 \frac{i_T \pi m}{N} + \Gamma i_T^2 \sin^2 \frac{\pi m}{N}} \left[G_1(m, n) \cos \frac{(i_T - 1)\pi m}{N} - G_2(m, n) \sin \frac{(i_T - 1)\pi m}{N} \right]$$

$$F_2(m, n) = \frac{i_T \sin \frac{\pi m}{N} \sin \frac{i_T \pi m}{N}}{\sin^2 \frac{i_T \pi m}{N} + \Gamma i_T^2 \sin^2 \frac{\pi m}{N}} \left[G_1(m, n) \sin \frac{(i_T - 1) \pi m}{N} + G_2(m, n) \cos \frac{(i_T - 1) \pi m}{N} \right]$$

对于 $m = 0$ 的情况, 有:

$$F_1(0, n) = \frac{G_1(0, n)}{1 + \Gamma}, \text{ 其中 } 0 \leq n \leq N-1$$

$$F_2(0, n) = \frac{G_2(0, n)}{1 + \Gamma}, \text{ 其中 } 0 \leq n \leq N-1$$

如果再对函数 $F_1(m, n)$ 和 $F_2(m, n)$ 做反傅里叶变换, 就可以得到如图6-9a所示的复原图像。我们可以把这幅图像与通过逆滤波得到的图6-7e和6-7f进行比较。



a) 对图6-7a (模糊图) 取 $\Gamma = 0.01$ 应用 Wiener 滤波

b) 对图6-8a (模糊且有高斯噪声 ($\sigma = 10$)) 取 $\Gamma = 0.5$ 应用 Wiener 滤波

c) 对图6-8c (模糊且高斯噪声有 ($\sigma = 20$)) 取 $\Gamma = 1$ 应用 Wiener 滤波

图6-9 用 Wiener 滤波复原的图像

图6-9b和6-9c所示的图像是受噪声污染的图6-8a和6-8c分别通过 Wiener 滤波得到的复原图像。这些图像可以分别和图6-8g和6-8i相比较。在所有的情况下, Wiener 滤波能得到了更好的结果。

6.27 假定退化过程是线性的, 为什么我们要用卷积定理而不是通过解一组线性方程来恢复结果呢

事实上, 我们要进行求解的一组线性方程是通过式(6-10), $\mathbf{g} = H\mathbf{f}$, 以矩阵的形式给出的。但是, 实际上含有噪声项的方程要更现实一些 (参见式(6-26)):

$$\mathbf{g} = H\mathbf{f} + \mathbf{v} \quad (6-75)$$

其中, \mathbf{v} 是向量形式的噪声项。

既然我们假设已经对这个退化过程的点扩散函数有所了解, 矩阵 H 就是假定已知的, 于是:

$$\mathbf{f} = H^{-1}\mathbf{g} - H^{-1}\mathbf{v} \quad (6-76)$$

其中 H 是一个 $N^2 \times N^2$ 的矩阵, \mathbf{f} , \mathbf{g} 和 \mathbf{v} 是 $N^2 \times 1$ 的向量。

6.28 式(6-76)看起来简单明了, 为什么还要那么麻烦地去用其他方法呢

式(6-76)主要存在两个问题:

1) 该方程对噪声是极端敏感的, 已经证明该方法只有在不可能达到的极低噪声水平下才有效。

2) 求解式(6-76)需要对一个 $N^2 \times N^2$ 的矩阵求逆, 而 N 一般取值为 500, 即使是用最先进的计算机进行计算也是非常困难的。

例6.14 说明逆矩阵复原对噪声的敏感性。

我们来考虑下面这个信号:

$$f(x) = 25 \sin \frac{2\pi x}{30}, \text{ 其中 } x = 0, 1, \dots, 29$$

假设该信号被某函数所模糊, 且该函数为每三个采样点加权以后再取平均。此过程可表示为: 用如下所示的矩阵 H 去乘上述离散信号从而得到一个模糊信号 $g(x)$ 。

$$H = \begin{pmatrix} 0.4 & 0.3 & 0 & 0 & \dots & 0 & 0 & 0 & 0.3 \\ 0.3 & 0.4 & 0.3 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0.3 & 0.4 & 0.3 & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0.3 & 0 & 0 & 0 & \dots & 0 & 0 & 0.4 & 0.3 \end{pmatrix}$$

为了引入一些噪声, 我们将 $g(x)$ 的元素近似取其最近的整数值。为了恢复初始信号, 我们可以用矩阵 H 的逆乘以这个模糊信号 $g(x)$ 。初始信号和复原后的信号如图6-10所示:

231

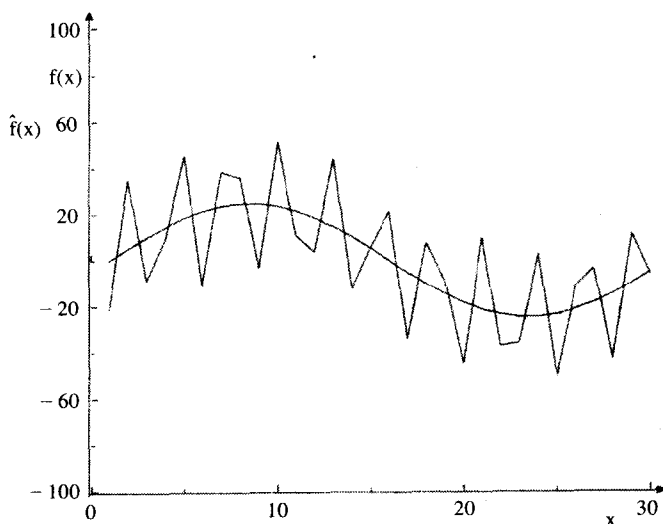


图6-10 一个初始信号和一个通过直接矩阵求逆复原的信号。信号中的噪声仅舍入误差产生

6.29 是否存在对矩阵H求逆的方法

存在, 矩阵 H 的逆很容易得到, 因为它是一个块循环矩阵。

6.30 什么矩阵是块循环的

矩阵 H 是块循环的, 如果它具有如下结构:

$$H = \begin{pmatrix} H_0 & H_{M-1} & H_{M-2} & \dots & H_1 \\ H_1 & H_0 & H_{M-1} & \dots & H_2 \\ H_2 & H_1 & H_0 & \dots & H_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ H_{M-1} & H_{M-2} & H_{M-3} & \dots & H_0 \end{pmatrix} \quad (6-77)$$

其中, H_0, H_1, \dots, H_{M-1} 是矩阵 H 的块, 并且它们自身都是循环矩阵。

6.31 什么矩阵是循环矩阵

矩阵 D 是循环矩阵的, 如果它具有如下结构:

$$D = \begin{pmatrix} d(0) & d(M-1) & d(M-2) & \dots & d(1) \\ d(1) & d(0) & d(M-1) & \dots & d(2) \\ d(2) & d(1) & d(0) & \dots & d(3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d(M-1) & d(M-2) & d(M-3) & \dots & d(0) \end{pmatrix} \quad (6-78)$$

在这样一个矩阵中, 它的每一列都是通过将其前一列所有元素向下移动一行, 并且前一列最下端的元素则为该列最上端的元素而得到的。

6.32 为什么块循环矩阵可以很容易地求逆

循环矩阵和块循环矩阵很容易求逆, 因为很容易找到它们的特征值和特征向量。

6.33 什么是循环矩阵的特征值和特征向量

我们定义下面的标量集:

$$\begin{aligned} \lambda(k) \equiv & d(0) + d(M-1) \exp \left[\frac{2\pi j}{M} k \right] + d(M-2) \exp \left[\frac{2\pi j}{M} 2k \right] \\ & + \dots + d(1) \exp \left[\frac{2\pi j}{M} (M-1)k \right] \end{aligned} \quad (6-79)$$

和向量集:

$$\mathbf{w}(k) = \begin{pmatrix} 1 \\ \exp\left[\frac{2\pi j}{M}k\right] \\ \exp\left[\frac{2\pi j}{M}2k\right] \\ \vdots \\ \exp\left[\frac{2\pi j}{M}(M-1)k\right] \end{pmatrix} \quad (6-80)$$

其中, k 的取值为 $k = 0, 1, 2, \dots, M-1$ 。直接置换可以证明有:

$$D\mathbf{w}(k) = \lambda(k)\mathbf{w}(k) \quad (6-81)$$

233

即, $\lambda(k)$ 是矩阵 D (式(6-78)所定义) 的特征值, $\mathbf{w}(k)$ 是其对应的特征向量。

6.34 已知矩阵特征值和特征向量如何求矩阵的逆

如果有矩阵 W , 它的列是矩阵 D 的特征向量, 则有:

$$D = W\Lambda W^{-1} \quad (6-82)$$

其中 W^{-1} 的元素为

$$W^{-1}(k, i) = \frac{1}{M} \exp\left[-j\frac{2\pi}{M}ki\right] \quad (6-83)$$

并且 Λ 是对角元素为其特征值的对角矩阵。因此, 矩阵 D 的逆就很容易得到:

$$D^{-1} = (W\Lambda W^{-1})^{-1} = (W^{-1})^{-1}\Lambda^{-1}W^{-1} = W\Lambda^{-1}W^{-1} \quad (6-84)$$

例6.15 考虑矩阵 W , 它的列 $\mathbf{w}(0), \mathbf{w}(1), \dots, \mathbf{w}(M-1)$ 由式(6-80)给出。证明具有元素:

$$Z(k, i) = \frac{1}{M} \exp\left(-\frac{2\pi j}{M}ki\right)$$

的矩阵 Z 是矩阵 W 的逆矩阵。

我们有:

$$W = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & e^{\frac{2\pi j}{M}} & e^{\frac{2\pi j}{M}2} & \dots & e^{\frac{2\pi j}{M}(M-1)} \\ 1 & e^{\frac{2\pi j}{M}2} & e^{\frac{2\pi j}{M}4} & \dots & e^{\frac{2\pi j}{M}2(M-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{\frac{2\pi j}{M}(M-1)} & e^{\frac{2\pi j}{M}2(M-1)} & \dots & e^{\frac{2\pi j}{M}(M-1)^2} \end{pmatrix}$$

$$Z = \frac{1}{M} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & e^{-\frac{2\pi j}{M}} & e^{-\frac{2\pi j}{M}2} & \dots & e^{-\frac{2\pi j}{M}(M-1)} \\ 1 & e^{-\frac{2\pi j}{M}2} & e^{-\frac{2\pi j}{M}4} & \dots & e^{-\frac{2\pi j}{M}2(M-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-\frac{2\pi j}{M}(M-1)} & e^{-\frac{2\pi j}{M}2(M-1)} & \dots & e^{-\frac{2\pi j}{M}(M-1)^2} \end{pmatrix}$$

$$ZW = \frac{1}{M} \begin{pmatrix} M & \sum_{k=0}^{M-1} e^{-\frac{2\pi j}{M}k} & \cdots & \sum_{k=0}^{M-1} e^{-\frac{2\pi j}{M}(M-1)k} \\ \sum_{k=0}^{M-1} e^{-\frac{2\pi j}{M}k} & M & \cdots & \sum_{k=0}^{M-1} e^{-\frac{2\pi j}{M}2(M-2)k} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{k=0}^{M-1} e^{-\frac{2\pi j}{M}(M-1)k} & \sum_{k=0}^{M-1} e^{-\frac{2\pi j}{M}2(M-2)k} & \cdots & M \end{pmatrix}$$

234

这个矩阵的所有非对角元素具有形式 $\sum_{k=0}^{M-1} e^{-\frac{2\pi j t k}{M}}$ 的形式, 其中 t 是正或负整数。这个和是一个首项为1且比率为 $q = e^{-\frac{2\pi j t}{M}}$ 的几何级数。应用公式:

$$\sum_{k=0}^{n-1} q^k = \frac{q^n - 1}{q - 1}$$

可以得到:

$$\sum_{k=0}^{M-1} e^{\frac{2\pi j t}{M}k} = \begin{cases} M & t = 0 \\ 0 & t \neq 0 \end{cases}$$

于是, $ZW = I$, 即 $Z = W^{-1}$ 。

例6.16 当 $M = 3$ 时, 由式(6-79)定义的 $\lambda(k)$ 和式(6-80)定义的 $\mathbf{w}(k)$ 分别是由式(6-78)定义的矩阵的特征值和特征向量, 其中 $k = 0, 1, 2$ 。

当 $M = 3$ 时, 重新定义矩阵 D 为:

$$D = \begin{pmatrix} d_0 & d_2 & d_1 \\ d_1 & d_0 & d_2 \\ d_2 & d_1 & d_0 \end{pmatrix}$$

而且有:

$$\mathbf{w}(k) = \begin{pmatrix} 1 \\ e^{\frac{2\pi j}{3}k} \\ e^{\frac{2\pi j}{3}2k} \end{pmatrix}, \text{ 其中 } k = 0, 1, 2$$

$$\lambda(k) = d_0 + d_2 e^{\frac{2\pi j}{3}k} + d_1 e^{\frac{2\pi j}{3}2k}, \text{ 其中 } k = 0, 1, 2$$

我们必须证明:

$$D\mathbf{w}(k) = \lambda(k)\mathbf{w}(k) \quad (6-85)$$

先来计算这个表达式的左端:

$$D\mathbf{w}(k) = \begin{pmatrix} d_0 & d_2 & d_1 \\ d_1 & d_0 & d_2 \\ d_2 & d_1 & d_0 \end{pmatrix} \begin{pmatrix} 1 \\ e^{\frac{2\pi j}{3}k} \\ e^{\frac{2\pi j}{3}2k} \end{pmatrix} = \begin{pmatrix} d_0 + d_2 e^{\frac{2\pi j}{3}k} + d_1 e^{\frac{2\pi j}{3}2k} \\ d_1 + d_0 e^{\frac{2\pi j}{3}k} + d_2 e^{\frac{2\pi j}{3}2k} \\ d_2 + d_1 e^{\frac{2\pi j}{3}k} + d_0 e^{\frac{2\pi j}{3}2k} \end{pmatrix} \quad (6-86)$$

235

再来计算式(6-85)的右端:

$$\lambda(k)\mathbf{w}(k) = \begin{pmatrix} d_0 + d_2 e^{\frac{2\pi j}{3}k} + d_1 e^{\frac{2\pi j}{3}2k} \\ d_0 e^{\frac{2\pi j}{3}k} + d_2 e^{\frac{4\pi j}{3}k} + d_1 e^{\frac{2\pi j}{3}3k} \\ d_0 e^{\frac{2\pi j}{3}2k} + d_2 e^{\frac{2\pi j}{3}3k} + d_1 e^{\frac{2\pi j}{3}4k} \end{pmatrix} \quad (6-87)$$

如果将式(6-86)和式(6-87)的右端矩阵的元素相比较, 并且考虑到:

$$e^{2\pi jk} = 1, \text{ 对于任意的整数 } k \text{ 成立}$$

和

$$e^{\frac{2\pi j}{3}4k} = e^{\frac{2\pi j}{3}3k} e^{\frac{2\pi j}{3}k} = e^{2\pi jk} e^{\frac{2\pi j}{3}k} = e^{\frac{2\pi j}{3}k}$$

可以看出式(6-85)是正确的。

例6.17 求下列矩阵的逆:

$$\begin{pmatrix} -1 & 0 & 2 & 3 \\ 3 & -1 & 0 & 2 \\ 2 & 3 & -1 & 0 \\ 0 & 2 & 3 & -1 \end{pmatrix}$$

在此例中, $M=4$, $d(0)=-1$, $d(1)=3$, $d(2)=2$, $d(3)=0$ 。于是:

$$\lambda(0) = -1 + 2 + 3 = 4 \Rightarrow \lambda(0)^{-1} = \frac{1}{4}$$

$$\lambda(1) = -1 + 2e^{\frac{2\pi j}{4}2} + 3e^{\frac{2\pi j}{4}3} = -1 - 2 - 3j = -3 - 3j \Rightarrow \lambda^{-1}(1) = \frac{-3+3j}{18} = \frac{-1+j}{6}$$

$$\lambda(2) = -1 + 2e^{\frac{2\pi j}{4}4} + 3e^{\frac{2\pi j}{4}6} = -1 + 2 - 3 = -2 \Rightarrow \lambda^{-1}(2) = -\frac{1}{2}$$

$$\lambda(3) = -1 + 2e^{\frac{2\pi j}{4}6} + 3e^{\frac{2\pi j}{4}9} = -1 - 2 + 3j = -3 + 3j \Rightarrow \lambda^{-1}(3) = \frac{-3-3j}{18} = \frac{-1-j}{6}$$

236

$$\mathbf{w}^T(0) = (1 \quad 1 \quad 1 \quad 1)$$

$$\mathbf{w}^T(1) = (1 \quad e^{\frac{2\pi j}{4}} \quad e^{\frac{2\pi j}{4}2} \quad e^{\frac{2\pi j}{4}3}) = (1 \quad j \quad -1 \quad -j)$$

$$\mathbf{w}^T(2) = (1 \quad e^{\frac{2\pi j}{4}2} \quad e^{\frac{2\pi j}{4}4} \quad e^{\frac{2\pi j}{4}6}) = (1 \quad -1 \quad 1 \quad -1)$$

$$\mathbf{w}^T(3) = (1 \quad e^{\frac{2\pi j}{4}3} \quad e^{\frac{2\pi j}{4}6} \quad e^{\frac{2\pi j}{4}9}) = (1 \quad -j \quad -1 \quad j)$$

我们利用这些向量可构造矩阵 W 和 W^{-1} , 然后应用式(6-84)即得:

$$\begin{aligned} D^{-1} &= \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{pmatrix} \begin{pmatrix} \frac{1}{4} & 0 & 0 & 0 \\ 0 & \frac{-1+j}{6} & 0 & 0 \\ 0 & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{-1-j}{6} \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix} \\ &= \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{pmatrix} \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{-1+j}{6} & \frac{1+j}{6} & \frac{1-j}{6} & \frac{-1-j}{6} \\ -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ \frac{-1-j}{6} & \frac{1-j}{6} & \frac{1+j}{6} & \frac{-1+j}{6} \end{pmatrix} \end{aligned}$$

$$= \frac{1}{4} \begin{pmatrix} -\frac{7}{12} & \frac{13}{12} & \frac{1}{12} & \frac{5}{12} \\ \frac{5}{12} & -\frac{7}{12} & \frac{13}{12} & \frac{1}{12} \\ \frac{1}{12} & \frac{13}{12} & -\frac{7}{12} & \frac{5}{12} \\ \frac{13}{12} & \frac{1}{12} & \frac{5}{12} & -\frac{7}{12} \end{pmatrix} = \frac{1}{48} \begin{pmatrix} -7 & 13 & 1 & 5 \\ 5 & -7 & 13 & 1 \\ 1 & 5 & -7 & 13 \\ 13 & 1 & 5 & -7 \end{pmatrix}$$

例6.18(B) 矩阵 W_N 的元素由下式给出:

$$W_N(k, n) = \frac{1}{\sqrt{N}} \exp \left(\frac{2\pi j}{N} kn \right)$$

其中 k 和 n 取值均为 $0, 1, 2, \dots, N-1$ 。其逆矩阵 W_N^{-1} (参见例6.10) 的元素由下式给出:

$$W_N^{-1}(k, n) = \frac{1}{\sqrt{N}} \exp \left(-\frac{2\pi j}{N} kn \right)$$

我们将矩阵 W 定义为矩阵 W_N 与其自身的Kronecker乘积, 证明矩阵 W 的逆矩阵则表示为矩阵 W_N^{-1} 与其自身的Kronecker乘积。

237

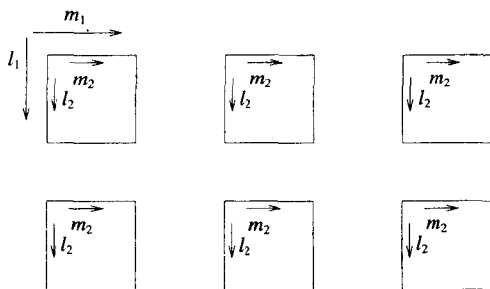
考虑矩阵 W 的一个元素 $W(m, l)$, 把整数 m 和 l 写成被 N 除所得商加余数的形式:

$$\begin{aligned} m &\equiv m_1 N + m_2 \\ l &\equiv l_1 N + l_2 \end{aligned}$$

由于 W 即为 $W_N \otimes W_N$, 我们有:

$$W(m, l) = \frac{1}{N} e^{(2\pi j/N) m_1 l_1} e^{(2\pi j/N) m_2 l_2} \quad (6-88)$$

指标 (m_1, l_1) 用来识别矩阵 W 的元素 $W(m, l)$ 所属的块, m_2 和 l_2 在每一个块中取遍所有可能的值。



同理, 我们可将矩阵 $Z \equiv W_N^{-1} \otimes W_N^{-1}$ 的元素写成:

$$Z(t, n) = \frac{1}{N} e^{-\frac{2\pi j}{N} t_1 n_1} e^{-\frac{2\pi j}{N} t_2 n_2} \quad (6-89)$$

则乘积矩阵 $A \equiv WZ$ 的元素如下所示:

$$\begin{aligned} A(k, n) &= \sum_{t=0}^{N^2-1} W(k, t) Z(t, n) \\ &= \frac{1}{N^2} \sum_{t=0}^{N^2-1} e^{\frac{2\pi j}{N} k_1 t_1} e^{\frac{2\pi j}{N} k_2 t_2} e^{-\frac{2\pi j}{N} t_1 n_1} e^{-\frac{2\pi j}{N} t_2 n_2} \\ &= \frac{1}{N^2} \sum_{t=0}^{N^2-1} e^{\frac{2\pi j}{N} (k_1 - n_1) t_1} e^{\frac{2\pi j}{N} (k_2 - n_2) t_2} \end{aligned}$$

如果将 t 写成 $t \equiv t_1 N + t_2$ 的形式, 那么基于 t 的和可表示成两部分, 一部分是基于 t_1 求和, 另一部分则是基于 t_2 求和:

$$A(k, n) = \frac{1}{N^2} \sum_{t_1=0}^{N-1} \left\{ e^{\frac{2\pi j}{N} (k_1 - n_1) t_1} \sum_{t_2=0}^{N-1} e^{\frac{2\pi j}{N} (k_2 - n_2) t_2} \right\} \quad (6-90)$$

由例6.10可知:

$$\sum_{m=0}^{M-1} e^{\frac{2\pi j m}{M} s} = \begin{cases} M & s = 0 \\ 0 & s \neq 0 \end{cases}$$

238

我们将上式应用到式(6-90)上, 先求里面的和, 然后再求外面的和:

$$\begin{aligned} A(k, n) &= \frac{1}{N^2} \sum_{t_1=0}^{N-1} e^{\frac{2\pi j}{N} (k_1 - n_1) t_1} \delta(k_2 - n_2) N \\ &= \frac{1}{N} \delta(k_2 - n_2) N \delta(k_1 - n_1) \\ &= \delta(k_2 - n_2) \delta(k_1 - n_1) \\ &= \delta(k - n) \end{aligned}$$

因此, 矩阵 A 的所有非对角元素均为0, 其对角元素均为1, 即矩阵 A 是单位矩阵。这就证明了由式(6-89) 给定元素的矩阵 Z 是由式(6-88) 给定元素的矩阵 W 的逆矩阵。

6.35 如何知道表达线性退化过程的矩阵 H 是块循环的

我们知道, 等式 $g = Hf$ 事实上等价于:

$$g(i, j) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} h(k, l, i, j) f(k, l) \quad (6-91)$$

其中, $h(k, l, i, j)$ 是该退化过程的点扩散函数。对于一个移不变的点扩散函数我们有:

$$g(i, j) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} f(k, l) h(i - k, j - l) \quad (6-92)$$

考虑分块矩阵 H 的一个块 (见式(6-77))。在每一个块中, l 和 j 的值保持不变, 也就是说在

每一个块中 $j-l$ 的值是常数。 $i-k$ 的值从 i 变为 $i-N+1$, 取遍其间所有的整数值。当 i 的值在下一行中增加1时, 所有 $i-k$ 的值均向右移动一个位置 (见式(1-26)和式(6-77))。所以, 矩阵 H 的每一个子矩阵可用 $j-l \equiv u$ 表示出来, 并有如下的循环形式:

$$H_u = \begin{pmatrix} h(0, u) & h(N-1, u) & h(N-2, u) & \dots & h(1, u) \\ h(1, u) & h(0, u) & h(N-1, u) & \dots & h(2, u) \\ h(2, u) & h(1, u) & h(0, u) & \dots & h(3, u) \\ \vdots & \vdots & \vdots & & \vdots \\ h(N-1, u) & h(N-2, u) & h(N-3, u) & \dots & h(0, u) \end{pmatrix} \quad (6-93)$$

注意, 这里我们假设 $h(v, u)$ 是周期函数, 每一个变量以 N 为周期。因此, 有 $h(1-N, u) = h((1-N) + N, u) = h(1, u)$ 等等。

故整个矩阵 H 可写成如下形式:

$$H = \begin{pmatrix} H_0 & H_{-1} & H_{-2} & \dots & H_{-M+1} \\ H_1 & H_0 & H_{-1} & \dots & H_{-M+2} \\ H_2 & H_1 & H_0 & \dots & H_{-M+3} \\ \vdots & \vdots & \vdots & & \vdots \\ H_{M-1} & H_{M-2} & H_{M-3} & \dots & H_0 \end{pmatrix} \quad (6-94)$$

其中, 由于 $h(v, u)$ 的周期性, 有 $H_{-1} = H_{M-1}$, $H_{-M+1} = H_1$ 等等。

6.36 如何对角化一个块循环矩阵

定义一个具有下列元素的矩阵,

$$W_N(k, n) = \frac{1}{\sqrt{N}} \exp \left[\frac{2\pi j}{N} kn \right] \quad (6-95)$$

并且有矩阵

$$W = W_N \otimes W_N \quad (6-96)$$

其中, \otimes 表示两个矩阵的Kronecker乘积。矩阵 $W_N(k, n)$ 的逆是一个具有下列元素的矩阵:

$$W_N^{-1}(k, n) = \frac{1}{\sqrt{N}} \exp \left[-\frac{2\pi j}{N} kn \right] \quad (6-97)$$

则矩阵 W 的逆可由下式给出 (参见例6.18):

$$W_N^{-1}(k, n) = W_N^{-1} \otimes W_N^{-1} \quad (6-98)$$

我们也可以定义一个如下形式的对角矩阵 Λ :

$$\Lambda(k, i) = \begin{cases} N\hat{H}(k_{\text{mod } N}, [\frac{k}{N}]) & \text{如果 } i = k \\ 0 & \text{如果 } i \neq k \end{cases} \quad (6-99)$$

其中, \hat{H} 是点扩散函数 h 的离散傅里叶变换:

$$\hat{H}(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} h(x, y) e^{-2\pi j(\frac{ux}{N} + \frac{vy}{N})} \quad (6-100)$$

通过直接矩阵乘积, 可以证明:

$$H = W \Lambda W^{-1} \quad (6-101)$$

由于矩阵 H 可以写成几个矩阵乘积的形式, 并且各分量的逆都很容易求得, 因此 H 的逆也很容易解得。

240

B6.5 式(6-101)如何证明?

首先必须找出矩阵 H 的一个元素 $H(f, g)$ 与点扩散函数 $h(x, y)$ 的关系。我们把下标 f, g 写成某块数与维数 N 的乘积再加上一个余数的形式:

$$\begin{aligned} f &\equiv f_1 N + f_2 \\ g &\equiv g_1 N + g_2 \end{aligned}$$

由于 f 和 g 可取遍从0到 N^2-1 之间的每一个可能的值, 我们可以形象地表示出矩阵 H 的 $N \times N$ 个块, 如下所示用下标 u 标出来:

$f_1 = 0$ $g_1 = 0$ $u = 0$	$f_1 = 0$ $g_1 = 1$ $u = -1$	$f_1 = 0$ $g_1 = 2$ $u = -2$...
$f_1 = 1$ $g_1 = 0$ $u = 1$	$f_1 = 1$ $g_1 = 1$ $u = 0$	$f_1 = 1$ $g_1 = 2$ $u = -1$...

我们注意到每个块都以下标 $f_1 - g_1$ 为特征, 并且在每一个块中, 由 $h(x, y)$ 生成的元素可以用变量值 $f_2 - g_2$ 表示, 我们可得出结论:

$$H(f, g) = h(f_2 - g_2, f_1 - g_1) \quad (6-102)$$

下面考虑矩阵 $W \Lambda W^{-1}$ 的元素:

$$A(m, n) \equiv \sum_{l=0}^{N^2-1} \sum_{t=0}^{N^2-1} W_{ml} \Lambda_{lt} W_{tn}^{-1}$$

由于 Λ 是对角矩阵, 基于 t 的求和只有在 $t = l$ 时才有效, 于是有:

$$A(m, n) = \sum_{l=0}^{N^2-1} W_{ml} \Lambda_{ll} W_{ln}^{-1}$$

由于 Λ_{ll} 是一个标量, 因此在被加数中可以改变其位置:

$$A(m, n) = \sum_{l=0}^{N^2-1} W_{ml} W_{ln}^{-1} \Lambda_{ll} \quad (6-103)$$

241

在方框B6.2中, 我们已证明了如何表示矩阵 W_{ml} 和 W_{ln}^{-1} 的元素, 如果将它们的指标写成被 N 除所得商和余数的形式:

$$\begin{aligned} m &\equiv Nm_1 + m_2 \\ l &\equiv Nl_1 + l_2 \\ n &\equiv Nn_1 + n_2 \end{aligned}$$

利用这些表达式, 以及由式(6-99)将 Λ_{ll} 定义为 $N\hat{H}(l_2, l_1)$, 可以得到:

$$A(m, n) = \sum_{l=0}^{N^2-1} e^{\frac{2\pi j}{N} m_1 l_1} e^{\frac{2\pi j}{N} m_2 l_2} \frac{1}{N^2} e^{-\frac{2\pi j}{N} l_1 n_1} e^{-\frac{2\pi j}{N} l_2 n_2} N\hat{H}(l_2, l_1)$$

重新调整一下, 有:

$$A(m, n) = \frac{1}{N} \sum_{l_1=0}^{N-1} \sum_{l_2=0}^{N-1} \hat{H}(l_2, l_1) e^{\frac{2\pi j}{N} (m_1 - n_1) l_1} e^{\frac{2\pi j}{N} (m_2 - n_2) l_2} \quad (6-104)$$

可以看出该表达式是 $h(m_2 - n_2, m_1 - n_1)$ 的傅里叶反变换, 因此有:

$$A(m, n) = h(m_2 - n_2, m_1 - n_1) \quad (6-105)$$

通过比较式(6-102)和式(6-105), 可以看到矩阵 H 和 $W\Lambda W^{-1}$ 具有同样的表达形式。

B6.6 什么是矩阵 H 的转置?

我们将证明 $H^T = W\Lambda^* W^{-1}$, 其中, Λ^* 是矩阵 Λ 的复共轭矩阵。

由方框B6.5中式(6-102)可知矩阵 H 的转置矩阵的元素可由下式给出:

$$H^T(f, g) = h(g_2 - f_2, g_1 - f_1) \quad (6-106)$$

矩阵 $W\Lambda^* W^{-1}$ 的元素 $A(m, n)$ 可由方框B6.5中类似式(6-104)给出, 但是由于 Λ^* 的元素被定义成式(6-100)的傅里叶变换 $\hat{H}(u, v)$ 的复共轭项形式, 因此 $A(m, n)$ 不是含有 $\hat{H}(l_2, l_1)$ 因子, 而是含有 $\hat{H}(-l_2, -l_1)$ 因子:

$$A(m, n) = \frac{1}{N} \sum_{l_1=0}^{N-1} \sum_{l_2=0}^{N-1} \hat{H}(-l_2, -l_1) e^{\frac{2\pi j}{N} (m_1 - n_1) l_1} e^{\frac{2\pi j}{N} (m_2 - n_2) l_2}$$

242

我们将上式中的求和变量置换为:

$$\tilde{l}_1 \equiv -l_1 \text{ 和 } \tilde{l}_2 \equiv -l_2$$

于是有:

$$A(m, n) = \frac{1}{N} \sum_{\tilde{l}_1=0}^{-N+1} \sum_{\tilde{l}_2=0}^{-N+1} \hat{H}(\tilde{l}_2, \tilde{l}_1) e^{\frac{2\pi j}{N} (-m_1 + n_1) \tilde{l}_1} e^{\frac{2\pi j}{N} (-m_2 + n_2) \tilde{l}_2}$$

由于我们是在一个周期上处理周期函数的求和问题, 因此可以不必管求和的范围, 只要对 N 个连贯的下标求和就行了, 至于是在哪个范围内求和都不会有什么影响。于是可写成:

$$A(m, n) = \frac{1}{N} \sum_{\tilde{l}_1=0}^{N-1} \sum_{\tilde{l}_2=0}^{N-1} \hat{H}(\tilde{l}_2, \tilde{l}_1) e^{\frac{2\pi j}{N}(-m_1+n_1)\tilde{l}_1} e^{\frac{2\pi j}{N}(-m_2+n_2)\tilde{l}_2}$$

可以看到上式的右边实质是基于 (n_2-m_2, n_1-m_1) 的 $\hat{H}(\tilde{l}_2, \tilde{l}_1)$ 的傅里叶反变换, 即:

$$A(m, n) = h(n_2 - m_2, n_1 - m_1) \quad (6-107)$$

通过直接与式(6-106)比较, 可以证明矩阵 H^T 和 $W\Lambda^*W^{-1}$ 是相等的。

- 例6.19 证明拉普拉斯算子 (Laplacian), 即, 一幅离散图像在像素点 (i, j) 处的二阶导数之和可由下式表达:

$$\Delta^2 f(i, j) = f(i-1, j) + f(i, j-1) + f(i+1, j) + f(i, j+1) - 4f(i, j)$$

在两相邻像素中间位置 $(i+\frac{1}{2}, j)$, 图像函数沿 i 轴的一阶导数可由一阶差分近似给出:

$$\Delta_i f(i+\frac{1}{2}, j) = f(i+1, j) - f(i, j) \quad [24]$$

同理, 在 $(i-\frac{1}{2}, j)$ 处沿 i 轴的一阶差分为:

$$\Delta_i f(i-\frac{1}{2}, j) = f(i, j) - f(i-1, j)$$

那么, 图像函数在 (i, j) 处沿 i 轴的二阶导数可以用 $(i+\frac{1}{2}, j)$ 和 $(i-\frac{1}{2}, j)$ 处的一阶差分的一阶差分近似得到, 即:

$$\begin{aligned} \Delta_i^2 f(i, j) &= \Delta_i f(i+\frac{1}{2}, j) - \Delta_i f(i-\frac{1}{2}, j) \\ &= f(i+1, j) - 2f(i, j) + f(i-1, j) \end{aligned} \quad (6-108)$$

同理, 图像函数在 (i, j) 处沿 j 轴的二阶导数可如下近似表示:

$$\begin{aligned} \Delta_j^2 f(i, j) &= \Delta_j f(i, j+\frac{1}{2}) - \Delta_j f(i, j-\frac{1}{2}) \\ &= f(i, j+1) - 2f(i, j) + f(i, j-1) \end{aligned} \quad (6-109)$$

将式(6-108)和式(6-109)两边相加即得结论。

例6.20 我们来考虑一个用列向量 \mathbf{f} 描述的 3×3 的图像。如果用一个 9×9 的矩阵 L 与向量 \mathbf{f} 相乘, 结果将会得到一个向量, 该向量在各位置上返回拉普拉斯算子的估算值。假设图像 f 在各方向上是以3为周期的周期函数, 那么矩阵 L 是什么类型的?

由例6.19, 可知在各个位置返回拉普拉斯算子估算值的矩阵算子是:

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

为了避免边缘影响, 我们首先将图像在所有方向上周期性延拓:

$$\begin{matrix} & & f_{31} & f_{32} & f_{33} & & \\ f_{13} & \begin{pmatrix} f_{11} & f_{12} & f_{13} \end{pmatrix} & & & f_{11} \\ f_{21} & \begin{pmatrix} f_{21} & f_{22} & f_{23} \end{pmatrix} & & & f_{21} \\ f_{33} & \begin{pmatrix} f_{31} & f_{32} & f_{33} \end{pmatrix} & & & f_{31} \\ & f_{11} & f_{12} & f_{13} & & \end{matrix}$$

244

通过观察拉普拉斯算子在一个像素点的作用值, 并将它作为权值, 可以构造下图所示的 9×9 的矩阵, 用该矩阵与列向量 \mathbf{f} 相乘, 即:

$$\begin{pmatrix} -4 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -4 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -4 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & -4 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & -4 \end{pmatrix} \begin{pmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{pmatrix}$$

容易看出, 该矩阵是一个以大小为 3×3 的块为单元的块循环矩阵。

例6.21(B) 利用例6.20中定义的矩阵估算如下图像的拉普拉斯算子:

$$\begin{pmatrix} 3 & 2 & 1 \\ 2 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

然后再利用例6.19中的公式估算上面图像的拉普拉斯算子。

$$\begin{pmatrix} -4 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -4 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -4 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & -4 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & -4 \end{pmatrix} \begin{pmatrix} 3 \\ 2 \\ 0 \\ 2 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -7 \\ -4 \\ 6 \\ -4 \\ 5 \\ 3 \\ 3 \\ 0 \\ -2 \end{pmatrix}.$$

245

如果我们要利用该公式, 必须先延拓图像的边缘像素点:

$$\begin{array}{ccccc} & & 0 & 0 & 1 \\ & & 1 & \left(\begin{array}{ccc} 3 & 2 & 1 \\ 2 & 0 & 1 \\ 0 & 0 & 1 \end{array} \right) & \begin{array}{c} 3 \\ 2 \\ 0 \end{array} \\ & & 3 & 2 & 1 \end{array}$$

拉普拉斯算子是:

$$\begin{pmatrix} 1+2+2-4 \times 3 & 3+1-4 \times 2 & 1+2+1+3-4 \times 1 \\ 3+1-4 \times 2 & 2+2+1 & 1+1+2-4 \times 1 \\ 2+1+3 & 2+1 & 1+1-4 \times 1 \end{pmatrix} = \begin{pmatrix} -7 & -4 & 3 \\ -4 & 5 & 0 \\ 6 & 3 & -2 \end{pmatrix}$$

注意, 不论我们利用局部计算公式还是利用矩阵乘法, 都得到同样的结果。

例6.22(B) 计算例6.20的矩阵的特征向量和特征值。

例6.20的矩阵 L 是:

$$\begin{pmatrix} -4 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -4 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -4 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & -4 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & -4 \end{pmatrix}$$

可以很容易地看出该矩阵是个以 3×3 的块为一个单元的块循环矩阵。要想找到它的特征向量, 首先要利用式(6-80), 取 $M=3$ 来定义向量 \mathbf{w} 。

$$\mathbf{w}(0) = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \mathbf{w}(1) = \begin{pmatrix} 1 \\ e^{\frac{2\pi j}{3}} \\ e^{\frac{4\pi j}{3}} \end{pmatrix} \quad \mathbf{w}(2) = \begin{pmatrix} 1 \\ e^{\frac{4\pi j}{3}} \\ e^{\frac{8\pi j}{3}} \end{pmatrix}$$

246

用这些向量作为列构成一个矩阵。我们再将该矩阵除以 $\sqrt{M} = \sqrt{3}$ 来标准化这个矩阵, 这样就构造出如式(6-95)所定义的矩阵, 即:

$$W_3 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 1 & 1 \\ 1 & e^{\frac{2\pi j}{3}} & e^{\frac{4\pi j}{3}} \\ 1 & e^{\frac{4\pi j}{3}} & e^{\frac{8\pi j}{3}} \end{pmatrix}$$

我们用上面这个矩阵与其自身的Kronecker乘积生成如式(6-96)所定义的矩阵 W :

$$W = \frac{1}{3} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & e^{\frac{2\pi j}{3}} & e^{\frac{4\pi j}{3}} & 1 & e^{\frac{2\pi j}{3}} & e^{\frac{4\pi j}{3}} & 1 & e^{\frac{2\pi j}{3}} & e^{\frac{4\pi j}{3}} \\ 1 & e^{\frac{4\pi j}{3}} & e^{\frac{8\pi j}{3}} & 1 & e^{\frac{4\pi j}{3}} & e^{\frac{8\pi j}{3}} & 1 & e^{\frac{4\pi j}{3}} & e^{\frac{8\pi j}{3}} \\ 1 & 1 & 1 & e^{\frac{2\pi j}{3}} & e^{\frac{2\pi j}{3}} & e^{\frac{2\pi j}{3}} & e^{\frac{4\pi j}{3}} & e^{\frac{4\pi j}{3}} & e^{\frac{4\pi j}{3}} \\ 1 & e^{\frac{2\pi j}{3}} & e^{\frac{4\pi j}{3}} & e^{\frac{2\pi j}{3}} & e^{\frac{4\pi j}{3}} & e^{\frac{6\pi j}{3}} & e^{\frac{4\pi j}{3}} & e^{\frac{6\pi j}{3}} & e^{\frac{8\pi j}{3}} \\ 1 & e^{\frac{4\pi j}{3}} & e^{\frac{8\pi j}{3}} & e^{\frac{2\pi j}{3}} & e^{\frac{6\pi j}{3}} & e^{\frac{10\pi j}{3}} & e^{\frac{4\pi j}{3}} & e^{\frac{8\pi j}{3}} & e^{\frac{12\pi j}{3}} \\ 1 & 1 & 1 & e^{\frac{4\pi j}{3}} & e^{\frac{4\pi j}{3}} & e^{\frac{4\pi j}{3}} & e^{\frac{8\pi j}{3}} & e^{\frac{8\pi j}{3}} & e^{\frac{8\pi j}{3}} \\ 1 & e^{\frac{2\pi j}{3}} & e^{\frac{4\pi j}{3}} & e^{\frac{4\pi j}{3}} & e^{\frac{6\pi j}{3}} & e^{\frac{8\pi j}{3}} & e^{\frac{8\pi j}{3}} & e^{\frac{10\pi j}{3}} & e^{\frac{12\pi j}{3}} \\ 1 & e^{\frac{4\pi j}{3}} & e^{\frac{8\pi j}{3}} & e^{\frac{4\pi j}{3}} & e^{\frac{8\pi j}{3}} & e^{\frac{12\pi j}{3}} & e^{\frac{8\pi j}{3}} & e^{\frac{12\pi j}{3}} & e^{\frac{16\pi j}{3}} \end{pmatrix}$$

该矩阵的列即是矩阵 L 的特征向量。注意这些特征向量对于具有相同结构的所有块循环矩阵都是一样的，并不依赖于具体的特征值。利用式(6-97)可构造矩阵 W 的逆矩阵，亦即矩阵 W 的复共轭。(注意，对于一般的酉矩阵，必须通过取其转置矩阵的复共轭得到其逆矩阵。这里由于 W 是一个对称矩阵，与其转置是相同的，故可通过直接取其复共轭得到其逆矩阵。)

$$W^{-1} = \frac{1}{3} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{4\pi j}{3}} & 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{4\pi j}{3}} & 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{4\pi j}{3}} \\ 1 & e^{-\frac{4\pi j}{3}} & e^{-\frac{8\pi j}{3}} & 1 & e^{-\frac{4\pi j}{3}} & e^{-\frac{8\pi j}{3}} & 1 & e^{-\frac{4\pi j}{3}} & e^{-\frac{8\pi j}{3}} \\ 1 & 1 & 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} & e^{-\frac{4\pi j}{3}} & e^{-\frac{4\pi j}{3}} & e^{-\frac{4\pi j}{3}} \\ 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{4\pi j}{3}} & e^{-\frac{2\pi j}{3}} & e^{-\frac{4\pi j}{3}} & e^{-\frac{6\pi j}{3}} & e^{-\frac{4\pi j}{3}} & e^{-\frac{6\pi j}{3}} & e^{-\frac{8\pi j}{3}} \\ 1 & e^{-\frac{4\pi j}{3}} & e^{-\frac{8\pi j}{3}} & e^{-\frac{2\pi j}{3}} & e^{-\frac{6\pi j}{3}} & e^{-\frac{10\pi j}{3}} & e^{-\frac{4\pi j}{3}} & e^{-\frac{8\pi j}{3}} & e^{-\frac{12\pi j}{3}} \\ 1 & 1 & 1 & e^{-\frac{4\pi j}{3}} & e^{-\frac{4\pi j}{3}} & e^{-\frac{4\pi j}{3}} & e^{-\frac{8\pi j}{3}} & e^{-\frac{8\pi j}{3}} & e^{-\frac{8\pi j}{3}} \\ 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{4\pi j}{3}} & e^{-\frac{4\pi j}{3}} & e^{-\frac{6\pi j}{3}} & e^{-\frac{8\pi j}{3}} & e^{-\frac{8\pi j}{3}} & e^{-\frac{10\pi j}{3}} & e^{-\frac{12\pi j}{3}} \\ 1 & e^{-\frac{4\pi j}{3}} & e^{-\frac{8\pi j}{3}} & e^{-\frac{4\pi j}{3}} & e^{-\frac{8\pi j}{3}} & e^{-\frac{12\pi j}{3}} & e^{-\frac{8\pi j}{3}} & e^{-\frac{12\pi j}{3}} & e^{-\frac{16\pi j}{3}} \end{pmatrix}$$

利用式(6-99)，可以通过计算矩阵 L 的傅里叶变换求其特征值。但是，首先我们需要确定用矩阵 L 描述的算子的核 $l(x, y)$ ，并利用式(6-100)求出 $l(x, y)$ 的傅里叶变换 $\hat{L}(u, v)$ 。

247

由例6.20可知核函数是：

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

那么，对于离散函数 $l(x, y)$ 我们可以确定其值为：

$$\begin{aligned} l(0, 0) &= -4, l(-1, -1) = 0, l(-1, 0) = 1, l(-1, 1) = 0 \\ l(0, -1) &= 1, l(0, 1) = 1, l(1, -1) = 0, l(1, 0) = 1, l(1, 1) = 0 \end{aligned}$$

注意，由于在式(6-100)中需假定所定义的函数 $h(x, y)$ 的变量 x 和 y 都是正值，因此上面这些值不能由式(6-100)直接求出。我们需要对核函数做一个平移变换，即把值-4放在新的核矩阵的左上角。通过读取矩阵 L 的第一列并进行卷绕，我们可以得到这样一个 3×3 的矩阵：

$$\begin{pmatrix} -4 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

于是有:

$$\begin{aligned} l(0,0) &= -4, l(0,1) = 1, l(0,2) = 1, l(1,0) = 1 \\ l(2,0) &= 1, l(1,1) = 0, l(1,2) = 0, l(2,1) = 0, l(2,2) = 0 \end{aligned}$$

将这些值用于式(6-100)可得:

$$\hat{L}(u,v) = \frac{1}{3} \left[-4 + e^{-\frac{2\pi j}{3}v} + e^{-\frac{2\pi j}{3}2v} + e^{-\frac{2\pi j}{3}u} + e^{-\frac{2\pi j}{3}2u} \right] \quad (6-110)$$

式(6-99)中所指的矩阵 L 的特征值,即出现在矩阵 $\Lambda(k,i)$ 的对角线上的值,是当 $u = k_{mod3}$ 且 $v = [\frac{k}{3}]$ 时的傅里叶变换 $\hat{L}(u,v)$ 的值,其中, $k = 0, 1, \dots, 8$ 。这些值可以利用式(6-110)计算得到:

$$\begin{aligned} \hat{L}(0,0) &= 0 \\ \hat{L}(0,1) &= \frac{1}{3} \left[-4 + e^{-\frac{2\pi j}{3}} + e^{-\frac{4\pi j}{3}} + 1 + 1 \right] = \frac{1}{3} [-2 - 2 \cos 60^\circ] = -1 \\ \hat{L}(0,2) &= \frac{1}{3} \left[-4 + e^{-\frac{4\pi j}{3}} + e^{-\frac{8\pi j}{3}} + 2 \right] = \frac{1}{3} [-2 + e^{-\frac{4\pi j}{3}} + e^{-\frac{2\pi j}{3}}] = -1 \\ \hat{L}(1,0) &= \hat{L}(0,1) = -1 \\ \hat{L}(1,1) &= \frac{1}{3} \left[-4 + 2e^{-\frac{2\pi j}{3}} + 2e^{-\frac{4\pi j}{3}} \right] = \frac{1}{3} [-4 - 4 \cos 60^\circ] = -2 \\ \hat{L}(1,2) &= \frac{1}{3} \left[-4 + e^{-\frac{4\pi j}{3}} + e^{-\frac{8\pi j}{3}} + e^{-\frac{2\pi j}{3}} + e^{-\frac{4\pi j}{3}} \right] = -2 \\ \hat{L}(2,0) &= \hat{L}(0,2) = -1 \\ \hat{L}(2,1) &= \hat{L}(1,2) = -2 \\ \hat{L}(2,2) &= \frac{1}{3} \left[-4 + 2e^{-\frac{4\pi j}{3}} + 2e^{-\frac{8\pi j}{3}} \right] = -2 \end{aligned}$$

248

其中,我们用到了下面的式子:

$$\begin{aligned} e^{-\frac{2\pi j}{3}} &= -\cos 60^\circ - j \sin 60^\circ = -\frac{1}{2} - j \frac{\sqrt{3}}{2} \\ e^{-\frac{4\pi j}{3}} &= -\cos 60^\circ + j \sin 60^\circ = -\frac{1}{2} + j \frac{\sqrt{3}}{2} \\ e^{-\frac{6\pi j}{3}} &= 1 \\ e^{-\frac{8\pi j}{3}} &= e^{-\frac{2\pi j}{3}} = -\cos 60^\circ - j \sin 60^\circ = -\frac{1}{2} - j \frac{\sqrt{3}}{2} \end{aligned}$$

由于矩阵 L 的第一个特征值是0,因此,矩阵 L 是奇异的。尽管我们可以利用式(6-101)将其对角化,但是我们不能通过对这个对角矩阵取逆来求矩阵 L 的逆。当矩阵 L 表示为一幅图像的拉普拉斯算子时并不值得奇怪,因为我们知道仅靠拉普拉斯算子的知识是无法复原原图像的。对矩阵 L 应用式(6-99)可以定义矩阵 Λ 为:

$$\Lambda = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6 \end{bmatrix}$$

利用已经定义的矩阵 W , W^{-1} 和 Λ , 我们可将 L 写成:

$$L = W\Lambda W^{-1}$$

249

上面这个等式可以通过直接相乘来验证。首先我们来计算矩阵 ΛW^{-1} :

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -e^{-\frac{2\pi j}{3}} & -e^{-\frac{4\pi j}{3}} & -1 & -e^{-\frac{2\pi j}{3}} & -e^{-\frac{4\pi j}{3}} & -1 & -e^{-\frac{2\pi j}{3}} & -e^{-\frac{4\pi j}{3}} \\ -1 & -e^{-\frac{4\pi j}{3}} & -e^{-\frac{8\pi j}{3}} & -1 & -e^{-\frac{4\pi j}{3}} & -e^{-\frac{8\pi j}{3}} & -1 & -e^{-\frac{4\pi j}{3}} & -e^{-\frac{8\pi j}{3}} \\ -1 & -1 & -1 & -e^{-\frac{2\pi j}{3}} & -e^{-\frac{2\pi j}{3}} & -e^{-\frac{2\pi j}{3}} & -e^{-\frac{4\pi j}{3}} & -e^{-\frac{4\pi j}{3}} & -e^{-\frac{4\pi j}{3}} \\ -2 & -2e^{-\frac{2\pi j}{3}} & -2e^{-\frac{4\pi j}{3}} & -2e^{-\frac{2\pi j}{3}} & -2e^{-\frac{4\pi j}{3}} & -2e^{-\frac{6\pi j}{3}} & -2e^{-\frac{4\pi j}{3}} & -2e^{-\frac{6\pi j}{3}} & -2e^{-\frac{8\pi j}{3}} \\ -2 & -2e^{-\frac{4\pi j}{3}} & -2e^{-\frac{8\pi j}{3}} & -2e^{-\frac{2\pi j}{3}} & -2e^{-\frac{6\pi j}{3}} & -2e^{-\frac{10\pi j}{3}} & -2e^{-\frac{4\pi j}{3}} & -2e^{-\frac{8\pi j}{3}} & -2e^{-\frac{12\pi j}{3}} \\ -1 & -1 & -1 & -e^{-\frac{4\pi j}{3}} & -e^{-\frac{4\pi j}{3}} & -e^{-\frac{4\pi j}{3}} & -e^{-\frac{8\pi j}{3}} & -e^{-\frac{8\pi j}{3}} & -e^{-\frac{8\pi j}{3}} \\ -2 & -2e^{-\frac{2\pi j}{3}} & -2e^{-\frac{4\pi j}{3}} & -2e^{-\frac{4\pi j}{3}} & -2e^{-\frac{6\pi j}{3}} & -2e^{-\frac{8\pi j}{3}} & -2e^{-\frac{10\pi j}{3}} & -2e^{-\frac{12\pi j}{3}} & -2e^{-\frac{14\pi j}{3}} \\ -2 & -2e^{-\frac{4\pi j}{3}} & -2e^{-\frac{8\pi j}{3}} & -2e^{-\frac{4\pi j}{3}} & -2e^{-\frac{8\pi j}{3}} & -2e^{-\frac{12\pi j}{3}} & -2e^{-\frac{8\pi j}{3}} & -2e^{-\frac{12\pi j}{3}} & -2e^{-\frac{16\pi j}{3}} \end{bmatrix}$$

如果我们考虑到:

$$\begin{aligned} e^{-\frac{10\pi j}{3}} &= e^{-\frac{4\pi j}{3}} = -\cos 60^\circ + j \sin 60^\circ = -\frac{1}{2} + j\frac{\sqrt{3}}{2} \\ e^{-\frac{12\pi j}{3}} &= 1 \\ e^{-\frac{16\pi j}{3}} &= e^{-\frac{4\pi j}{3}} = -\cos 60^\circ + j \sin 60^\circ = -\frac{1}{2} + j\frac{\sqrt{3}}{2} \end{aligned}$$

并且将矩阵 W 左乘上面这个矩阵, 我们就可复原矩阵 L .

•
•
•

6.37 现在我们知道了如何解决对 H 求逆的问题, 但是如何解决式(6-76)对噪声的极端敏感问题

我们可以通过对该问题的解附加一个平滑约束, 使得它不至于波动太大。也就是说, 我们希望重构图像的二阶导数总体上比较小。在每个像素点, 图像沿着每个轴的二阶导数之和可以近似得出:

$$\Delta^2 f(i, k) = f(i-1, k) + f(i, k-1) + f(i+1, k) + f(i, k+1) - 4f(i, k) \quad (6-111)$$

这是在 (i, k) 处的拉普拉斯算子的值。我们选择附加的平滑约束是为了使得在每个像素点处的拉普拉斯算子的值的平方和最小:

$$\sum_{k=1}^N \sum_{i=1}^N [\Delta^2 f(i, k)]^2 = \text{最小} \quad (6-112)$$

每个像素点处的拉普拉斯算子的值, 都可由拉普拉斯算子 (其形式为 $N^2 \times N^2$ 矩阵) 作用于列向量 \mathbf{f} (其大小为 $N^2 \times 1$) 所产生的 $L\mathbf{f}$ 计算得出。 $L\mathbf{f}$ 是一个向量, 它的所有元素的平方和可由 $(L\mathbf{f})^T L\mathbf{f}$ 计算得到。这个约束就是:

$$(L\mathbf{f})^T L\mathbf{f} = \text{最小} \quad (6-113) \quad \boxed{250}$$

6.38 如何利用矩阵逆的约束

我们再次用矩阵的方程式来表述求解 \mathbf{f} 的问题:

$$\mathbf{g} = H\mathbf{f} + \mathbf{v} \quad (6-114)$$

假定噪声向量 \mathbf{v} 是未知的, 但是它的某些统计特征是知道的, 例如, 我们知道:

$$\mathbf{v}^T \mathbf{v} = \varepsilon \quad (6-115)$$

ε 这个量跟噪声的方差有关, 并且可以仅利用图像自身中均匀亮度的区域估算得到。如果我们利用式(6-114)求出向量 \mathbf{v} , 代入式(6-115)中, 就有:

$$(\mathbf{g} - H\mathbf{f})^T (\mathbf{g} - H\mathbf{f}) = \varepsilon \quad (6-116)$$

于是, 问题就转化为在式(6-116)的约束下最小化式(6-113)。该问题的解是傅里叶变换形式的滤波器:

$$\hat{M}(u, v) = \frac{1}{N} \frac{\hat{H}^*(u, v)}{|\hat{H}(u, v)|^2 + \gamma |\hat{L}(u, v)|^2} \quad (6-117)$$

通过分子和分母同乘以 $\hat{H}(u, v)$, 我们可以提取复原滤波器的传递函数:

$$\hat{M}(u, v) = \frac{1}{N} \frac{1}{\hat{H}(u, v)} \frac{|\hat{H}(u, v)|^2}{|\hat{H}(u, v)|^2 + \gamma |\hat{L}(u, v)|^2} \quad (6-118)$$

其中, γ 是一个常数, $\hat{L}(u, v)$ 是一个 $N \times N$ 的矩阵 L 的傅里叶变换, 并具有下列性质: 如果我们用它左乘一幅图像, 将会输出一个和该图像一样大小的阵列, 它的值是相应像素点处的拉普拉斯算子的估算值。参数 γ 的作用是在平滑输出与数据关注之间取得平衡。

B6.7 找出问题的解: 最小化 $(L\mathbf{f})^T L\mathbf{f}$, 约束如下:

$$[\mathbf{g} - H\mathbf{f}]^T [\mathbf{g} - H\mathbf{f}] = \varepsilon \quad (6-119)$$

利用拉格朗日乘数 (Lagrange multiply) (见方框B6.9) 的方法, 方程的解必须满足:

$$\frac{\partial}{\partial \mathbf{f}} [\mathbf{f}^T L^T L \mathbf{f} + \lambda (\mathbf{g} - H\mathbf{f})^T (\mathbf{g} - H\mathbf{f})] = 0 \quad (6-120)$$

其中 λ 是一个常数。这是关于一个向量的导数, 它将产生一组含有 N^2 个方程 (一个方程对应向量 \mathbf{f} 的一个分量) 的方程组, 它和式(6-116)构成一组有 $N^2 + 1$ 个方程的方程组, 共有 $N^2 + 1$ 个未知量, 即向量 \mathbf{f} 的 N^2 个分量再加上 λ 。

如果 \mathbf{a} 和 \mathbf{b} 都分别是一个向量, 它就可以表示成 (参见例6.23):

$$\frac{\partial \mathbf{f}^T \mathbf{a}}{\partial \mathbf{f}} = \mathbf{a} \quad (6-121)$$

$$\frac{\partial \mathbf{b}^T \mathbf{f}}{\partial \mathbf{f}} = \mathbf{b} \quad (6-122)$$

并且, 如果 A 是一个 $N^2 \times N^2$ 的方阵, 那么 (参见例6.24):

$$\frac{\partial \mathbf{f}^T A \mathbf{f}}{\partial \mathbf{f}} = (A + A^T) \mathbf{f} \quad (6-123)$$

我们将式(6-121)、式(6-122)和式(6-123)应用到式(6-120)来计算这个导数:

$$\frac{\partial}{\partial \mathbf{f}} \left[\underbrace{\mathbf{f}^T (L^T L) \mathbf{f}}_{\substack{\text{式(6-123)} \\ \text{其中 } A = L^T L}} + \lambda (\mathbf{g}^T \mathbf{g} - \underbrace{\mathbf{g}^T H \mathbf{f}}_{\substack{\text{式(6-122)} \\ \text{其中 } \mathbf{b} = H^T \mathbf{g}}} - \underbrace{\mathbf{f}^T H^T \mathbf{g}}_{\substack{\text{式(6-121)} \\ \text{其中 } \mathbf{a} = H^T \mathbf{g}}} + \underbrace{\mathbf{f}^T H^T H \mathbf{f}}_{\substack{\text{式(6-123)其中} \\ A = H^T H}}) \right] = 0$$

$$\Rightarrow (2L^T L) \mathbf{f} + \lambda (-H^T \mathbf{g} - H^T \mathbf{g} + 2H^T H \mathbf{f}) = 0$$

$$\Rightarrow (H^T H + \gamma L^T L) \mathbf{f} = H^T \mathbf{g} \quad (6-124)$$

其中 $\gamma = \frac{1}{\lambda}$ 。对于块循环矩阵式(6-124)很容易求解, 于是有:

$$\mathbf{f} = [H^T H + \gamma L^T L]^{-1} H^T \mathbf{g} \quad (6-125)$$

γ 可以从式(6-119)中求出。

B6.8 求解式(6-124)

由于 H 和 L 都是块循环矩阵（参见例6.20、例6.22和方框B6.6），它们可以被写成：

$$\begin{aligned} H &= W\Lambda_h W^{-1} & H^T &= W\Lambda_h^* W^{-1} \\ L &= W\Lambda_l W^{-1} & L^T &= W\Lambda_l^* W^{-1} \end{aligned} \quad (6-126)$$

于是

$$\begin{aligned} H^T H + \gamma L^T L &= W\Lambda_h^* W^{-1} W\Lambda_h W^{-1} + \gamma W\Lambda_l^* W^{-1} W\Lambda_l W^{-1} \\ &= W\Lambda_h^* \Lambda_h W^{-1} + \gamma W\Lambda_l^* \Lambda_l W^{-1} \\ &= W(\Lambda_h^* \Lambda_h + \gamma \Lambda_l^* \Lambda_l) W^{-1} \end{aligned} \quad (6-127)$$

我们将式(6-126)和式(6-127)代到式(6-124)中，即可得到：

$$W(\Lambda_h^* \Lambda_h + \gamma \Lambda_l^* \Lambda_l) W^{-1} \mathbf{f} = W\Lambda_h^* W^{-1} \mathbf{g} \quad (6-128)$$

首先在上式两边同时左乘 W^{-1} ，可得：

$$(\Lambda_h^* \Lambda_h + \gamma \Lambda_l^* \Lambda_l) W^{-1} \mathbf{f} = \Lambda_h^* W^{-1} \mathbf{g} \quad (6-129)$$

注意当 Λ_h^* 、 $\Lambda_h^* \Lambda_h$ 和 $\Lambda_l^* \Lambda_l$ 都是对角矩阵时，这个等式就表示向量 $W^{-1} \mathbf{f}$ 和 $W^{-1} \mathbf{g}$ 的相关元素之间一一对应的关系。

利用例6.26的结果，我们可以写成：

$$\Lambda_h^* \Lambda_h = N^2 |\hat{H}(u, v)|^2 \text{ 和 } \Lambda_l^* \Lambda_l = N^2 |\hat{L}(u, v)|^2$$

其中 $|\hat{L}(u, v)|^2$ 是矩阵 L 的傅里叶变换。然后利用例6.25的结果，可以写成：

$$W^{-1} \mathbf{f} = \hat{F}(u, v) \text{ 和 } W^{-1} \mathbf{g} = \hat{G}(u, v)$$

最后，将 Λ_h^* 用式(6-99)中所定义的式子代替，则式(6-129)就变成：

$$\begin{aligned} N^2 \left[|\hat{H}(u, v)|^2 + \gamma |\hat{L}(u, v)|^2 \right] \hat{F}(u, v) &= N \hat{H}^*(u, v) \hat{G}(u, v) \Rightarrow \\ \hat{F}(u, v) &= \frac{1}{N} \frac{\hat{H}^*(u, v) \hat{G}(u, v)}{|\hat{H}(u, v)|^2 + \gamma |\hat{L}(u, v)|^2} \end{aligned} \quad (6-130)$$

253

6.39 Wiener滤波器和加约束的矩阵逆滤波器之间的关系是怎样的

这两种滤波器看起来相似（参见式(6-47)和式(6-118)），但是它们有很多的不同之处：

1) Wiener滤波器的最优复原设计是基于大量相似图像样本的平均统计意义的，而加约束的矩阵逆滤波器仅仅处理一幅图像并附加约束下求解。

2) Wiener滤波器基于的假设是已知频谱密度的随机场是齐次的。而加约束的矩阵逆滤波器的假设是我们仅仅知道噪声的某些统计性质。

在加约束的矩阵复原方法中,不同的滤波器构造可以通过对同一个公式简单地改变平滑标准来得到。例如,一个标准可以是在所有位置最小化一阶导数的平方和,而不是二阶导数。与式(6-117)惟一的差别只在矩阵 L 上。

- **例6.23(B)** 对一个向量求导数,其实质是对这个向量的每个元素求导数。对于向量 \mathbf{a} , \mathbf{b} 和 \mathbf{f}
• 则有:

$$\frac{\partial \mathbf{f}^T \mathbf{a}}{\partial \mathbf{f}} = \mathbf{a} \text{ 和 } \frac{\partial \mathbf{f}^T \mathbf{b}}{\partial \mathbf{f}} = \mathbf{b}$$

假如向量 \mathbf{a} , \mathbf{b} 和 \mathbf{f} 都是 $N \times 1$ 的,那么,我们有

$$\mathbf{f}^T \mathbf{a} = (f_1 \quad f_2 \quad \dots \quad f_N) \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{pmatrix} = f_1 a_1 + f_2 a_2 + \dots + f_N a_N$$

直接代入就有

$$\frac{\partial \mathbf{f}^T \mathbf{a}}{\partial \mathbf{f}} \equiv \begin{pmatrix} \frac{\partial \mathbf{f}^T \mathbf{a}}{\partial f_1} \\ \frac{\partial \mathbf{f}^T \mathbf{a}}{\partial f_2} \\ \vdots \\ \frac{\partial \mathbf{f}^T \mathbf{a}}{\partial f_N} \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{pmatrix} \Rightarrow \frac{\partial \mathbf{f}^T \mathbf{a}}{\partial \mathbf{f}} = \mathbf{a}$$

同理,

$$\mathbf{b}^T \mathbf{f} = b_1 f_1 + b_2 f_2 + \dots + b_N f_N$$

则

$$\frac{\partial \mathbf{b}^T \mathbf{f}}{\partial \mathbf{f}} = \begin{pmatrix} \frac{\partial \mathbf{b}^T \mathbf{f}}{\partial f_1} \\ \frac{\partial \mathbf{b}^T \mathbf{f}}{\partial f_2} \\ \vdots \\ \frac{\partial \mathbf{b}^T \mathbf{f}}{\partial f_N} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{pmatrix} \Rightarrow \frac{\partial \mathbf{b}^T \mathbf{f}}{\partial \mathbf{f}} = \mathbf{b}$$

例6.24(B) 如果 \mathbf{f} 是一个 $N \times 1$ 的向量, A 是一个 $N \times N$ 的矩阵,试证明:

$$\frac{\partial \mathbf{f}^T A \mathbf{f}}{\partial \mathbf{f}} = (A + A^T) \mathbf{f}$$

利用例6.23的结果我们可以很容易看到:

$$\begin{aligned}
 \frac{\partial \mathbf{f}^T \mathbf{A} \mathbf{f}}{\partial \mathbf{f}} &= \frac{\partial \mathbf{f}^T (\mathbf{A} \mathbf{f})}{\partial \mathbf{f}} + \frac{\partial (\mathbf{f}^T \mathbf{A}) \mathbf{f}}{\partial \mathbf{f}} \\
 &= \mathbf{A} \mathbf{f} + \frac{\partial (\mathbf{A}^T \mathbf{f})^T \mathbf{f}}{\partial \mathbf{f}} \\
 &= \mathbf{A} \mathbf{f} + \mathbf{A}^T \mathbf{f} = (\mathbf{A} + \mathbf{A}^T) \mathbf{f}
 \end{aligned}$$

(这里我们利用了 $\mathbf{A} \mathbf{f}$ 和 $\mathbf{A}^T \mathbf{f}$ 都是向量的事实。)

例6.25(B) 如果 \mathbf{g} 是与一幅 3×3 的图像 G 相对应的列向量, W^{-1} 是例6.18中当 $N = 3$ 时所定义的矩阵, 则向量 $W^{-1} \mathbf{g}$ 是图像 G 的离散傅里叶变换 \hat{G} 。

假设

$$\mathbf{G} = \begin{pmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{pmatrix} \text{ 并且 } W_3^{-1} = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 1 & 1 \\ 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} \\ 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} \end{pmatrix}$$

255

那么

$$\begin{aligned}
 W^{-1} &= W_3^{-1} \otimes W_3^{-1} \\
 &= \frac{1}{3} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} \\ 1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & 1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & 1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} \\ 1 & 1 & 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} \\ 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}3} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}3} & e^{-\frac{2\pi j}{3}4} \\ 1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}3} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}4} & e^{-\frac{2\pi j}{3}3} \\ 1 & 1 & 1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} \\ 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}3} & e^{-\frac{2\pi j}{3}4} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}3} \\ 1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}4} & e^{-\frac{2\pi j}{3}3} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}3} & e^{-\frac{2\pi j}{3}2} \end{pmatrix}
 \end{aligned}$$

如果我们应用 $e^{-\frac{2\pi j}{3}3} = e^{-2\pi j} = 1$ 和 $e^{-\frac{2\pi j}{3}4} = e^{-\frac{2\pi j}{3}} e^{-\frac{2\pi j}{3}} = e^{-\frac{2\pi j}{3}}$, 那么矩阵就可以简化, 于是可以得到:

$$\begin{aligned}
 W^{-1} \mathbf{g} &= \\
 &= \frac{1}{3} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} \\ 1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & 1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & 1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} \\ 1 & 1 & 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} \\ 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & 1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} \\ 1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}3} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} \\ 1 & 1 & 1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}} \\ 1 & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}3} & e^{-\frac{2\pi j}{3}4} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}3} \\ 1 & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}2} & e^{-\frac{2\pi j}{3}4} & e^{-\frac{2\pi j}{3}3} & e^{-\frac{2\pi j}{3}} & e^{-\frac{2\pi j}{3}3} & e^{-\frac{2\pi j}{3}2} \end{pmatrix} \begin{pmatrix} g_{11} \\ g_{21} \\ g_{31} \\ g_{12} \\ g_{22} \\ g_{32} \\ g_{13} \\ g_{23} \\ g_{33} \end{pmatrix} \\
 &= \frac{1}{3} \begin{pmatrix} g_{11}+g_{21}+g_{31}+g_{12}+g_{22}+g_{32}+g_{13}+g_{23}+g_{33} \\ g_{11}+g_{21}e^{-\frac{2\pi j}{3}}+g_{31}e^{-\frac{2\pi j}{3}2}+g_{12}+g_{22}e^{-\frac{2\pi j}{3}}+g_{32}e^{-\frac{2\pi j}{3}2}+g_{13}+g_{23}e^{-\frac{2\pi j}{3}}+g_{33}e^{-\frac{2\pi j}{3}2} \\ \vdots \\ g_{11}+g_{21}+g_{31}+g_{12}e^{-\frac{2\pi j}{3}}+g_{22}e^{-\frac{2\pi j}{3}2}+g_{32}e^{-\frac{2\pi j}{3}2}+g_{13}e^{-\frac{2\pi j}{3}}+g_{23}e^{-\frac{2\pi j}{3}}+g_{33}e^{-\frac{2\pi j}{3}} \\ \vdots \end{pmatrix}
 \end{aligned}$$

[256]

仔细分析该向量的所有元素, 我们发现它们其实是 G 的傅里叶变换在频率 (u, v) 取不同值时的分量, 其中 $u = 0, 1, 2$, 而 $v = 0, 1, 2$, 即:

$$\begin{pmatrix} \hat{G}(0,0) \\ \hat{G}(1,0) \\ \hat{G}(2,0) \\ \hat{G}(0,1) \\ \hat{G}(1,1) \\ \hat{G}(2,1) \\ \hat{G}(0,2) \\ \hat{G}(1,2) \\ \hat{G}(2,2) \end{pmatrix}$$

这就证明了向量 $W^{-1}\mathbf{g}$ 以列商量的形式生成了 G 的傅里叶变换。

例6.26(B) 证明如果 Λ 是如式(6-99)定义的矩阵, 那么 $\Lambda^*\Lambda$ 是一个对角矩阵, 并且沿对角线其第 k 个元素值是 $N^2 |\hat{H}(k_2, k_1)|^2$, 其中 $k_2 \equiv k_{\text{mod } N}$, $k_1 \equiv \lfloor \frac{k}{N} \rfloor$ 。

由式(6-99) Λ 的定义, 可以写成:

$$\Lambda = \begin{pmatrix} N\hat{H}(0,0) & 0 & 0 & \dots & 0 \\ 0 & N\hat{H}(1,0) & 0 & \dots & 0 \\ 0 & 0 & N\hat{H}(2,0) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & N\hat{H}(N-1, N-1) \end{pmatrix}$$

那么

$$\Lambda^* = \begin{pmatrix} N\hat{H}^*(0,0) & 0 & 0 & \dots & 0 \\ 0 & N\hat{H}^*(1,0) & 0 & \dots & 0 \\ 0 & 0 & N\hat{H}^*(2,0) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & N\hat{H}^*(N-1, N-1) \end{pmatrix}$$

[257]

显然

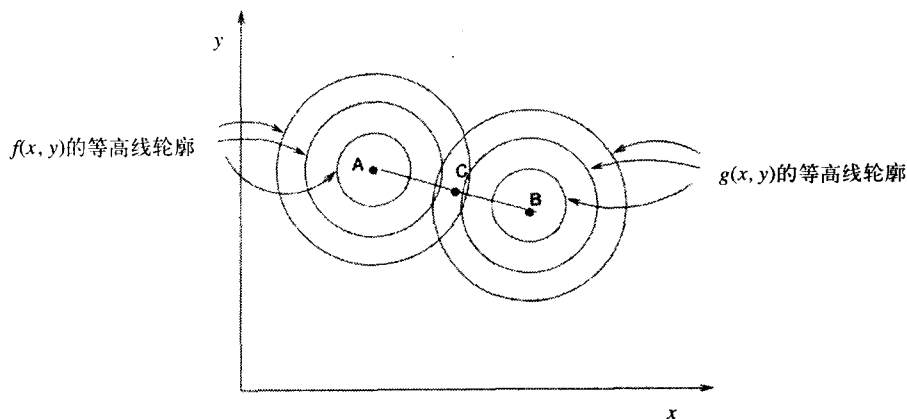
$$\Lambda^*\Lambda = \begin{pmatrix} N|\hat{H}(0,0)|^2 & 0 & 0 & \dots & 0 \\ 0 & N|\hat{H}(1,0)|^2 & 0 & \dots & 0 \\ 0 & 0 & N|\hat{H}(2,0)|^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & N|\hat{H}(N-1, N-1)|^2 \end{pmatrix} \quad \vdots$$

B6.9 为什么拉格朗日乘数方法是可行的?

假定我们希望同时满足下面两个方程:

$$\begin{aligned} f(x, y) &= 0 \\ g(x, y) &= 0 \end{aligned} \quad (6-131)$$

我们假设在 (x, y) 平面中的 A 点满足式(6-131)中第一个方程, 在 B 点满足第二个方程, 那么对于同一个 (x, y) 值不可能同时满足上面两个方程。我们希望在平面上找到一个 C 点, 使该点是不满足上面两方程的最小折衷点。该点的位置取决于当分别离开 A 点和 B 点时, $f(x, y)$ 和 $g(x, y)$ 的值从0开始变化的快慢。



我们来分别考虑围绕 A 点和 B 点的等高轮廓线。当轮廓线越远离 A 点时, 函数 $|f(x, y)|$ 的值越大; 同样当轮廓线越远离 B 点时, 函数 $|g(x, y)|$ 的值也越大。使 $|f(x, y)|$ 和 $|g(x, y)|$ 的值都尽可能小 (最小偏差的约束要求是 $|f(x, y)| = |g(x, y)| = 0$) 的 C 点, 它一定在 A 的某条轮廓线与 B 点某条轮廓线刚好接触且两线没有相交的位置。当这两条曲线刚好相接触时, 它们的切线是平行的。假设沿曲线 $f = \text{常数}$ 的切向量为 ∇f , 沿曲线 $g = \text{常数}$ 的切向量为 ∇g 。这两个切向量必须最大程度地满足约束条件, 它们不需要有相等的数量级, 而只需要有相同的方向就够了。因此, 我们说 C 点是由方程 $\nabla f = \mu \nabla g$ 的解, 其中 μ 是由两个向量的不同数量级所确定的常数。也就是说, 同时满足式(6-131)给出的两个方程的解实际上就是下列微分方程的解:

$$\nabla f + \lambda \nabla g = 0$$

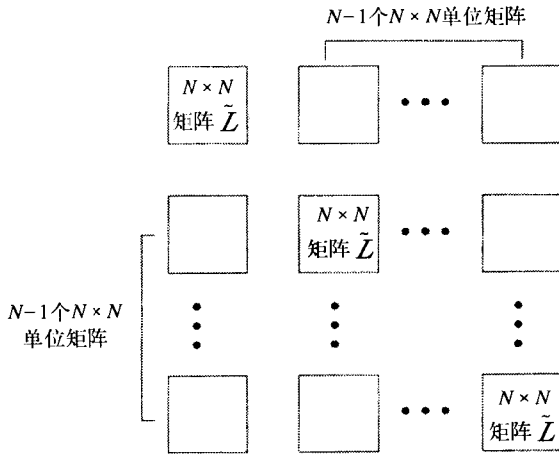
其中 λ 是拉格朗日乘数, 即一个任意的常数。

258

• 例6.27 通过复原一幅运动模糊的图像, 说明在实际中如何应用约束矩阵的逆。

• 为了将式(6-117)所给出的滤波器应用到图6-7a中的模糊图像, 我们必须首先定义表示约束的矩阵 $\hat{L}(u, v)$ 。

根据例6.19的步骤, 我们可以看到矩阵 $L(i, j)$ 是具有如下所示结构的一个 $N^2 \times N^2$ 的矩阵, 为了得到每个位置的拉普拉斯算子的值, 我们必须用矩阵 $L(i, j)$ 乘以一个 $N \times N$ 的图像。



矩阵 \tilde{L} 具有下面的形式:

$$\tilde{L} = \begin{pmatrix} -4 & 1 & \overbrace{0 \ 0 \ \dots \ 0}^{N-3 \text{ 个 } 0} & 1 \\ 1 & -4 & 1 & 0 \ \dots \ 0 \\ 0 & 1 & -4 & 1 \ \dots \ 0 \\ 0 & 0 & 1 & -4 \ \dots \ 0 \\ 0 & 0 & 0 & 1 \ \dots \ 0 \\ \vdots & \vdots & \vdots & \vdots \ \dots \ \vdots \\ 0 & 0 & 0 & 0 \ \dots \ -4 & 1 \\ 1 & 0 & 0 & 0 \ \dots \ 1 & -4 \end{pmatrix}$$

为了构造所需要的核, 我们必须取矩阵 L 的第一列, 做卷绕操作形成一个 $N \times N$ 的矩阵。矩阵 L 的第一列包括矩阵 \tilde{L} 的第一列元素(N 个元素)和 $N-1$ 个大小为 $N \times N$ 的单位矩阵的第一列。我们必须把这 N^2 个元素按列顺序写成一个 $N \times N$ 的矩阵 L' , 即

$$L' = \begin{pmatrix} -4 & 1 & 1 & \dots & 1 \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & \dots & 0 \end{pmatrix}$$

它就是出现在约束矩阵逆滤波器中的矩阵的傅里叶变换。

现在我们来利用式(6-100)计算当 $N = 128$ 时的傅里叶变换。假设这个变换的实部和虚部分别是 $L_1(m, n)$ 和 $L_2(m, n)$, 于是

$$|\hat{L}(m, n)|^2 = L_1^2(m, n) + L_2^2(m, n)$$

将传递函数(6-33)代入到式(6-117)中就可以获得我们将要用到的滤波器的传递函数, 即:

$$\hat{M}(m, n) = \frac{1}{N} \frac{\frac{1}{i_T \sin \frac{\pi m}{N}} \sin \frac{i_T \pi m}{N}}{\sin^2 \frac{i_T \pi m}{N} + \gamma (L_1^2(m, n) + L_2^2(m, n))} e^{j \frac{\pi m}{N} (i_T - 1)}$$

注意当 $m = 0$ 时, 我们需用下列公式:

$$\hat{M}(0, n) = \frac{1}{N} \frac{1}{1 + \gamma (L_1^2(0, n) + L_2^2(0, n))}, \text{ 其中 } 0 \leq n \leq N-1$$

就像在Wiener滤波情形所做的那样, 我们可以计算出原图像的傅里叶变换的实部和虚部分别为:

$$F_1(m, n) = \frac{1}{N} \frac{i_T \sin \frac{\pi m}{N} \sin \frac{i_T \pi m}{N} \left[G_1(m, n) \cos \frac{(i_T - 1) \pi m}{N} - G_2(m, n) \sin \frac{(i_T - 1) \pi m}{N} \right]}{\sin^2 \frac{i_T \pi m}{N} + \gamma (L_1^2(m, n) + L_2^2(m, n)) i_T^2 \sin^2 \frac{\pi m}{N}}$$

$$F_2(m, n) = \frac{1}{N} \frac{i_T \sin \frac{\pi m}{N} \sin \frac{i_T \pi m}{N} \left[G_1(m, n) \sin \frac{(i_T - 1) \pi m}{N} + G_2(m, n) \cos \frac{(i_T - 1) \pi m}{N} \right]}{\sin^2 \frac{i_T \pi m}{N} + \gamma (L_1^2(m, n) + L_2^2(m, n)) i_T^2 \sin^2 \frac{\pi m}{N}}$$

当 $0 < m \leq N-1$ 且 $0 < n \leq N-1$ 时这些等式是成立的。但是当 $m = 0$ 时, 我们需用下列等式:

$$F_1(0, n) = \frac{1}{N} \frac{G_1(0, n)}{1 + \gamma (L_1^2(0, n) + L_2^2(0, n))}$$

$$F_2(0, n) = \frac{1}{N} \frac{G_2(0, n)}{1 + \gamma (L_1^2(0, n) + L_2^2(0, n))}$$

如果我们对函数 $F_1(m, n)$ 和 $F_2(m, n)$ 作为实部和虚部进行傅里叶逆变换, 就得到了复原后的图像。对图6-7b、6-8a和6-8c的复原结果在图6-11中给出。注意 γ 的不同取值, 即, 不同程度的平滑必须应用到噪声程度不同的图像中。

261



a) 对图6-7b(模糊图像)取 $\gamma = 0.001$, 通过对约束矩阵求逆复原

b) 对图6-8a (模糊且有高斯噪声($\sigma = 10$)) 取 $\gamma = 0.05$, 通过约束矩阵求逆复原

c) 对图6-8c (模糊且有高斯噪声($\sigma = 20$)), 取 $\gamma = 0.1$, 通过约束矩阵求逆复原

图6-11 用约束矩阵的逆的方法复原

6.40 本章要点

本章主要探讨了用来更正（即复原）图像的受损数据的几种方法。图像的复原问题要求知道关于原始信号或者图像场景的先验知识，在这一方面，图像的复原问题不同于图像的增强问题。图像的几何复原要求已知一些参考点的准确位置。

图像的灰度复原要求退化噪声、模糊过程和原始图像本身的某些统计性质的知识。通常，我们会忽略对原始图像本身统计性质的知识要求，而只是在解上附加一些空间平滑约束，这主要是基于“世界很大程度上是平滑的”这一启示的。对退化过程和未退化图像选择合理的模型后，就可以对这个复原原始图像值的问题求解。

当图像的复原问题是非线性时，该问题就变的很困难，但我们可以借助局部或者全局的最优方法来解决。不过，如果假设退化过程是线性移不变的，并且把要求的解仅仅限制在线性解范畴内，也可以找到形如卷积滤波器的简单解。对于同一幅沿水平方向运动模糊的图像，

图6-12归纳了通过不同复原方法得到的结果。



图 6-12

第7章 图像分割和边缘检测

7.1 本章概述

本章主要介绍一些图像处理的方法，利用这些方法，可以为机器视觉系统做好图像的预处理。这些方法包括图像分割（image segmentation）和图像的边缘检测（edge detection），它们的目的是从一幅图像中提取信息，使得输出图像所包含的数据要远远少于原始图像的数据，但是与丢掉的数据相比，这些少量数据所包含的信息却与机器视觉系统的模块有更大的相关性。

7.2 图像分割和边缘检测的目的是什么

图像分割和边缘检测的目的是在一幅图像中提取不同区域的轮廓；这就是，将图像分割成不同的区域，每个区域都是由大致相同的像素组成。例如，这些区域可能有相同亮度或者相同的颜色，说明这些区域属于同一个物体或者一个物体的一部分。

7.3 如何将一幅图像分成一些统一的区域

最简单的方法之一就是利用直方图和阈值（thresholding）。如果将具有特定灰度值的像素的个数和该灰度值描绘在坐标上，就可以得到该幅图像的直方图。对直方图适当地进行标准化，则直方图实质上表示某一个灰度级出现的概率密度函数。假定有一幅图背景是深色的，其中有一个浅色的物体，我们要从这幅图像中提取出该物体，对于这幅图像来说，它的直方图将有两个峰，在两峰之间有一个谷。

我们可以选择直方图的谷对应的灰度值作为阈值，在图7-1中用 t_0 来表示，并且将所有灰度值高于 t_0 的像素标记为物体像素，将所有灰度值低于 t_0 的像素标记为背景像素。

265

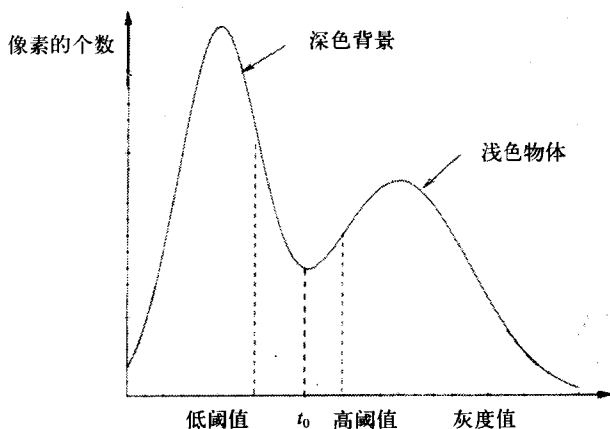


图7-1 在一幅深色背景中包含浅色物体的直方图

7.4 给图像加“标记”有什么意义

当我们说从一幅图像中提取出一个物体时，意思就是说把组成该物体的像素看成一样的。为了表达这种信息，我们给出一个与原图像同样大小的数组，给每个像素赋予一个标记。组成物体的所有像素有相同的标记，同样，组成图像背景的所有像素被赋予另一个标记。这个标记通常是一个数字，但也可以用其他符号，比如说一个字母或者是一种颜色。它本质上是一个名称，仅仅具有象征性的意义，因此，这些标记不能看作是数，我们不能用处理灰度图像的方法来处理标记图像。通常，标记图像可以看作表示了每个像素所属的类别的分类图像 (classified image)。

7.5 当直方图的谷不是很明显时如何处理

如果一幅图像的直方图的谷不是很明显时，这就意味着有些背景像素的灰度值与物体像素的灰度值相同。这些像素特别容易在物体的边缘附近出现，这些像素比较模糊，没有明显的定义。我们可以用滞后阈值法 (hysteresis thresholding) 来处理这些像素，所谓滞后阈值法是指我们在谷的每边选一个阈值 (见图7-1)，也就是选择两个阈值，而不是一个阈值。

两个阈值中的较大的被用来定义物体的“确实的中心”，较小的用于区分这些像素的位置上的相邻关系：只有当像素的灰度值大于较小的阈值和小于较大的阈值，并且与其相邻的像素确实是物体像素时，该像素才能标记为物体像素。

图7-2表示了深色背景下有一个浅色物体的图像，以及它的直方图。图7-2c显示了利用单阈值分割出来物体，同时直方图中标出了该阈值 t 的位置。图7-2d显示用两个阈值分割出来的物体，同时直方图中标出了这两个阈值 t_1 和 t_2 的位置。

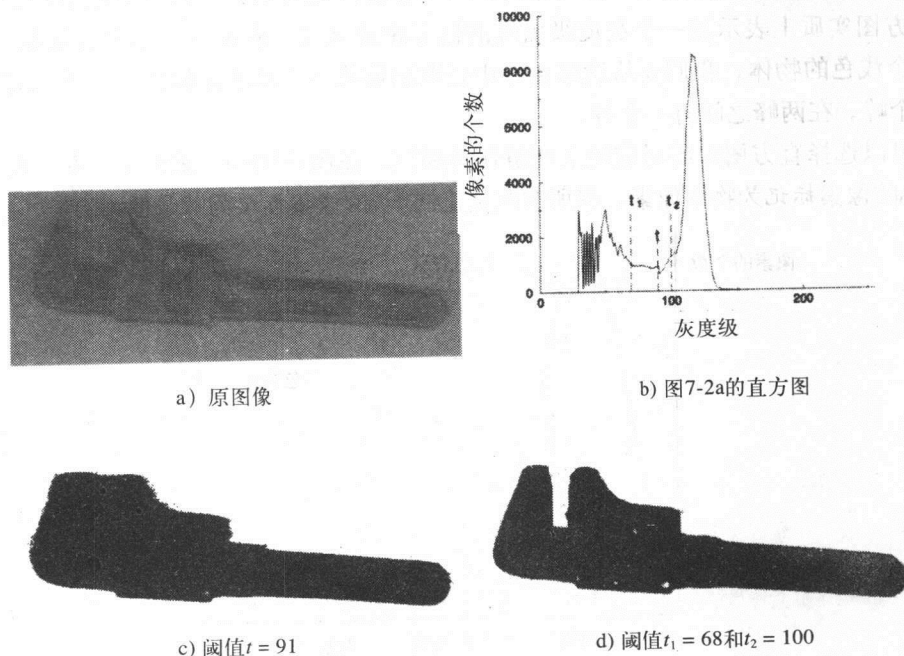


图7-2 单阈值法和滞后阈值法

总之, 我们尽可能用一种优化的方法选择全局阈值, 尽管我们知道采用这种方法一定会错分一些像素, 但可以尽可能使错分的像素个数最少。

7.6 如何最小化错分像素的数目

如果我们有关于物体和背景的像素灰度值分布的先验知识, 则可以使错分的像素数最小。例如, 如果我们知道物体在整个图像区域中所占的百分比是 θ , 那么, θ 就表示一个像素属于物体像素的先验概率。显然, 背景所占百分比为 $1-\theta$, 表示一个像素有 $1-\theta$ 的概率属于背景像素。我们可以选择阈值使得指定为物体的所有像素数与全部像素数的比刚好是 θ 。这种方法叫做 p -tile法, 进一步假设, 如果我们知道了属于物体的像素和属于背景的像素的灰度值的概率密度函数, 就可以选择合适的阈值使误差最小。

267

B7.1 关于带参数的积分的微分。

假设带参数的积分 $I(\lambda)$ (其中 λ 为参数) 定义如下:

$$I(\lambda) = \int_{a(\lambda)}^{b(\lambda)} f(x; \lambda) dx$$

关于参数 λ 的微分公式通过下式给出, 这就是著名的莱布尼兹法则 (Leibnitz rule):

$$\frac{dI(\lambda)}{d\lambda} = \frac{db(\lambda)}{d\lambda} f(b(\lambda); \lambda) - \frac{da(\lambda)}{d\lambda} f(a(\lambda); \lambda) + \int_{a(\lambda)}^{b(\lambda)} \frac{\partial f(x; \lambda)}{\partial \lambda} dx \quad (7-1)$$

7.7 如何选择最小误差阈值

假设我们知道物体和背景像素分布的概率密度函数分别为 $p_o(x)$ 和 $p_b(x)$ 。

如果选择的阈值为 t (见图7-3), 将一个物体像素当作背景像素进行分类时, 误分的概率为:

$$\int_{-\infty}^t p_o(x) dx$$

同样, 将一个背景像素当作物体像素进行分类时, 误分的概率为:

$$\int_t^{+\infty} p_b(x) dx$$

换句话说, 误分的整体概率是由在阈值 t 的两边的两个概率密度函数的误分产生的。进一步假设, 物体的像素所占的比例为 θ , 则图像背景的像素所占的比例为 $1-\theta$, 于是总的误差为:

268

$$E(t) = \theta \int_{-\infty}^t p_o(x) dx + (1 - \theta) \int_t^{+\infty} p_b(x) dx \quad (7-2)$$

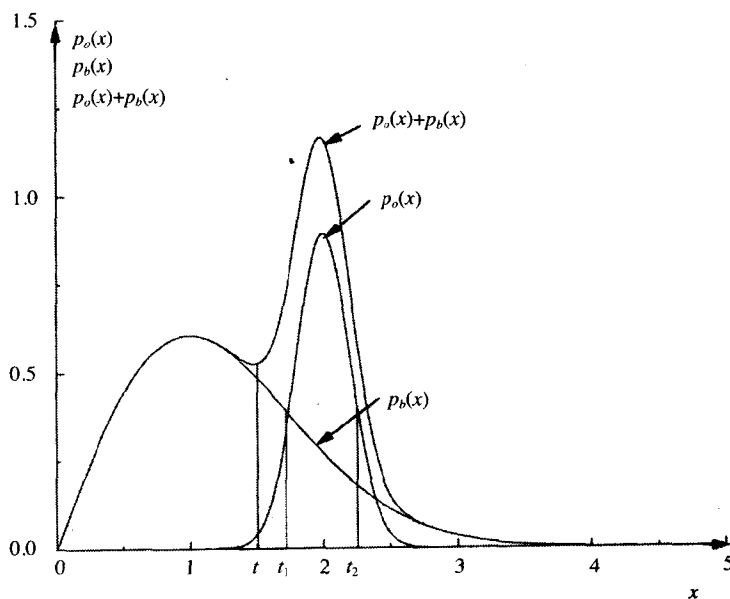


图7-3 物体像素值的概率密度函数($p_o(x)$)和图像背景像素值的概率密度函数($p_b(x)$)。

如果我们将一幅图像的直方图标准化,可以得到其积分和为1

选择 t 使得 $E(t)$ 最小,首先对 $E(t)$ 求关于 t 的微分(见方框B7.1),然后令它为零:

$$\begin{aligned} \frac{\partial E}{\partial t} &= \theta p_o(t) - (1 - \theta) p_b(t) = 0 \Rightarrow \\ \theta p_o(t) &= (1 - \theta) p_b(t) \end{aligned} \quad (7-3)$$

上述方程的解给出了两类像素总体分布下的最小误差阈值。

例7.1(B) 由式(7-2)推出式(7-3)。

我们利用式(7-1)给出的莱布尼兹法则对式(7-2)给出的 $E(t)$ 求微分,可以得到参数 λ 和 t 的对应关系。

对于第一个积分:

$$\begin{aligned} a(\lambda) &\rightarrow -\infty \quad (\text{一个常数, 其微分为零}) \\ b(\lambda) &\rightarrow t \\ f(x; \lambda) &\rightarrow p_o(x) \quad (\text{与要微分的参数 } t \text{ 无关}) \end{aligned}$$

对第二个积分:

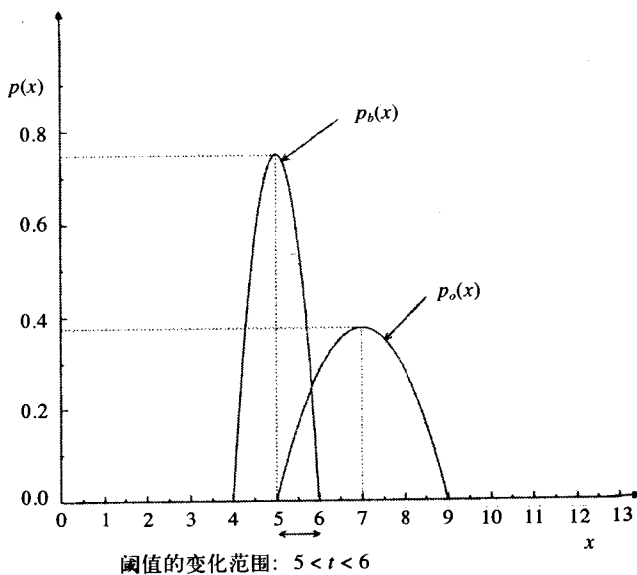
$$\begin{aligned} a(\lambda) &\rightarrow t \\ b(\lambda) &\rightarrow -\infty \quad (\text{一个常数, 其微分为零}) \\ f(x; \lambda) &\rightarrow p_b(x) \quad (\text{与参数 } t \text{ 无关}) \end{aligned}$$

于是,可以得到式(7-3)。

例7.2 图像中物体和背景像素灰度值的分布由以下的概率密度函数给出:

$$p(x) = \begin{cases} \frac{3}{4a^3} [a^2 - (x-b)^2] & b-a < x < b+a \\ 0 & \text{其他} \end{cases}$$

其中,对背景来说, $a=1, b=5$,对物体来说, $a=2, b=7$ 。在平面坐标上描绘出这两个分布的图形,并确定阈值的变化范围。



例7.3 假如图像中物体像素在整幅图像中所占的比例为 $\frac{8}{9}$, 对例7.2中的问题,试确定阈值,使得错分的像素最少。

将下式代入式(7-3)

$$\theta = \frac{8}{9} \Rightarrow 1 - \theta = \frac{1}{9}$$

$$p_b(t) = \frac{3}{4}(-t^2 - 24 + 10t) \quad p_o(t) = \frac{3}{32}(-t^2 - 45 + 14t)$$

可得:

$$\frac{1}{9} \frac{3}{4}(-t^2 - 24 + 10t) = \frac{8}{9} \frac{3}{32}(-t^2 - 45 + 14t) \Rightarrow$$

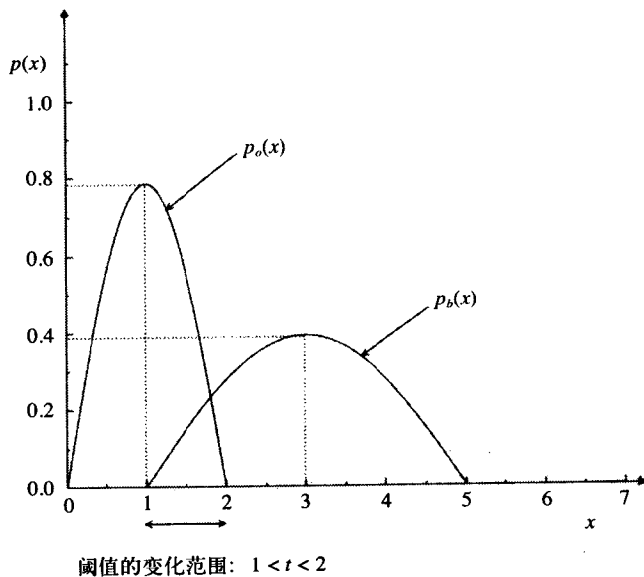
$$-24 + 10t = -45 + 14t \Rightarrow 4t = 21 \Rightarrow t = \frac{21}{4} = 5.25$$

(7-4)

例7.4 图像中物体和背景像素灰度值的分布由以下的概率密度函数给出:

$$p(x) = \begin{cases} \frac{\pi}{4a} \cos \frac{(x-x_0)\pi}{2a} & x_0 - a \leq x \leq x_0 + a \\ 0 & \text{其他} \end{cases}$$

其中, 对物体来说, $x_0 = 1, a = 1$, 对背景来说, $x_0 = 3, a = 2$ 。在平面坐标上描绘出这两个概率密度函数的分布图形。假如图像中物体像素占了总像素的三分之一, 试通过最佳阈值的方法, 确定物体像素被错分的百分数。



应用式(7-3)得:

$$\theta = \frac{1}{3} \Rightarrow 1 - \theta = \frac{2}{3}$$

$$p_o(x) = \frac{\pi}{4} \cos \frac{(x-1)\pi}{2} \quad p_b(x) = \frac{\pi}{8} \cos \frac{(x-3)\pi}{4}$$

式(7-3)变为:

$$\frac{1}{3} \frac{\pi}{4} \cos \frac{(t-1)\pi}{2} = \frac{2}{3} \frac{\pi}{8} \cos \frac{(t-3)\pi}{4} \Rightarrow$$

$$\Rightarrow \cos \frac{(t-1)\pi}{2} = \cos \frac{(t-3)\pi}{4} \Rightarrow$$

$$\Rightarrow \frac{(t-1)\pi}{2} = -\frac{(t-3)\pi}{4}$$

首先考虑 $\frac{(t-1)\pi}{2} = \frac{(t-3)\pi}{4} \Rightarrow 2t-2=t-3 \Rightarrow t=-1$ 。这个值超出了可以接受的阈值范围, 所以这个解没有意义, 于是:

$$\frac{(t-1)\pi}{2} = -\frac{(t-3)\pi}{4} \Rightarrow 2t-2 = -t+3 \Rightarrow 3t=5 \Rightarrow t = \frac{5}{3}$$

这就是最小误差下的阈值, 由图像中所有灰度值大于 $\frac{5}{3}$ 的物体像素确定了物体错分像素的比例。如果定义一个新的积分变量 $y = x-1$, 可得:

$$\begin{aligned} \int_{\frac{5}{3}}^2 \frac{\pi}{4} \cos \frac{(x-1)\pi}{2} dx &= \frac{\pi}{4} \int_{\frac{2}{3}}^1 \cos \frac{y\pi}{2} dy = \frac{\pi}{4} \left. \sin \frac{y\pi}{2} \right|_{\frac{2}{3}}^1 \\ &= \frac{1}{2} \left(\sin \frac{\pi}{2} - \sin \frac{\pi}{3} \right) = \frac{1}{2} (1 - \sin 60^\circ) = \frac{1}{2} \left(1 - \frac{\sqrt{3}}{2} \right) \\ &= \frac{2-1.7}{4} = \frac{0.3}{4} = 0.075 = 7.5\% \end{aligned}$$

•
•
•

7.8 当物体和背景像素是正态分布时的最小误差阈值是什么

假设图像中组成物体的像素服从正态分布, 其均值为 μ_o , 标准方差为 σ_o , 同样假设背景的像素也服从正态分布, 其均值为 μ_b , 标准方差为 σ_b :

$$\begin{aligned} p_o(x) &= \frac{1}{\sqrt{2\pi}\sigma_o} \exp \left[-\frac{(x-\mu_o)^2}{2\sigma_o^2} \right] \\ p_b(x) &= \frac{1}{\sqrt{2\pi}\sigma_b} \exp \left[-\frac{(x-\mu_b)^2}{2\sigma_b^2} \right] \end{aligned} \quad (7-5)$$

将上式代入式(7-3)可得:

$$\begin{aligned} \theta \frac{1}{\sqrt{2\pi}\sigma_o} \exp \left[-\frac{(t-\mu_o)^2}{2\sigma_o^2} \right] &= (1-\theta) \frac{1}{\sqrt{2\pi}\sigma_b} \exp \left[-\frac{(t-\mu_b)^2}{2\sigma_b^2} \right] \Rightarrow \\ \exp \left[-\frac{(t-\mu_o)^2}{2\sigma_o^2} + \frac{(t-\mu_b)^2}{2\sigma_b^2} \right] &= \frac{1-\theta}{\theta} \frac{\sigma_o}{\sigma_b} \Rightarrow \\ -\frac{(t-\mu_o)^2}{2\sigma_o^2} + \frac{(t-\mu_b)^2}{2\sigma_b^2} &= \ln \left[\frac{\sigma_o}{\sigma_b} \frac{1-\theta}{\theta} \right] \Rightarrow \\ (t^2 + \mu_b^2 - 2t\mu_b)\sigma_o^2 - (t^2 + \mu_o^2 - 2t\mu_o)\sigma_b^2 &= 2\sigma_o^2\sigma_b^2 \ln \left[\frac{\sigma_o}{\sigma_b} \frac{1-\theta}{\theta} \right] \Rightarrow \\ (\sigma_o^2 - \sigma_b^2)t^2 + 2(-\mu_b\sigma_o^2 + \mu_o\sigma_b^2)t + \mu_b^2\sigma_o^2 - \mu_o^2\sigma_b^2 - 2\sigma_o^2\sigma_b^2 \ln \left[\frac{\sigma_o}{\sigma_b} \frac{1-\theta}{\theta} \right] &= 0 \end{aligned} \quad (7-6)$$

273

这是关于 t 的二次方程。一般来说, 除非这两个正态分布有相同的标准方差, 否则这个方程有两个解。假设 $\sigma_o = \sigma_b$, 上述的表达式具有下列的形式:

$$2(\mu_o - \mu_b)\sigma_o^2 t + (\mu_b^2 - \mu_o^2)\sigma_o^2 - 2\sigma_o^4 \ln \left(\frac{1-\theta}{\theta} \right) = 0 \Rightarrow$$

$$t = \frac{\sigma_o^2}{\mu_o - \mu_b} \ln \left(\frac{1-\theta}{\theta} \right) - \frac{\mu_o + \mu_b}{2} \quad (7-7)$$

这就是误差最小下的阈值。

7.9 式(7-6)的两个解的意义是什么

当 $\sigma_o \neq \sigma_b$ 时, 式(7-6)的二次项不为零, 就存在两个阈值, 分别为 t_1 和 t_2 。可以证明这两个阈值分布在最陡的分布峰的两边。假设最陡的分布峰刚好反映的是物体像素的分布(见图7-3), 于是, 正确的阈值化应该是刚好将所有灰度值满足 $t_1 < x < t_2$ 的像素标记为物体像素。

274 第二个阈值表示较平坦的分布有一个长长的尾部, 灰度值满足 $x > t_2$ 的像素大多数情况下属于带有长长的尾部的较平坦的分布, 而不是属于较陡的分布。

• 例7.5 从图像7-4a得到一个最佳阈值并用它来阈值化这幅图像。

图7-4d显示出利用最佳阈值化方法对图7-2a所示的图像阈值化后的结果。首先, 直方图中两个主要的峰看作是等同的, 于是选择一个高斯分布与直方图左边的峰相匹配, 高斯分布的标准方差可以通过反复试验得到, 这样的话, 可以得到一个最佳的匹配。之所以首先选择左边的峰来匹配, 是因为这个峰比较平坦, 所以应该有最长的尾部, 而这个尾部对于其他分布的峰值有作用。一旦第一个峰匹配好之后, 从直方图减去用来匹配的高斯函数的值。假如相减后的结果是个负数, 就令这个结果为零。图7-4a显示了整个直方图, 图7-4b显示用高斯分布匹配第一个峰后的直方图, 其中高斯分布的均值和标准方差分别为 $\mu_o = 50$ 和 $\sigma_o = 7.5$ 。图7-4c显示除去高斯分布后剩下的直方图, 定义负值为零, 并用另一个高斯分布来匹配剩下的直方图, 其均值和标准方差分别为 $\mu_b = 117$ 和 $\sigma_b = 7$ 。两个高斯分布的振幅分别为 $A_o = 20477$ 和 $A_b = 56597$ 。通过积分这两个匹配函数, 可以估算出 θ 的值, 也就是物体像素所占的比例。

$$\frac{\theta}{1-\theta} = \frac{A_o \int_{-\infty}^{\infty} e^{-\frac{(x-\mu_o)^2}{2\sigma_o^2}} dx}{A_b \int_{-\infty}^{\infty} e^{-\frac{(x-\mu_b)^2}{2\sigma_b^2}} dx} = \frac{A_o \sqrt{2\pi}\sigma_o}{A_b \sqrt{2\pi}\sigma_b} = \frac{A_o\sigma_o}{A_b\sigma_b}$$

因此

$$\theta = \frac{A_o\sigma_o}{A_o\sigma_o + A_b\sigma_b}$$

在这个例子中, 我们估算出 $\theta = 0.272$, 将这值代入式(7-6)可得两个解分别为 $t_1 = -1213$ 和 $t_2 = 74$ 。图7-4d表示利用阈值 t_2 对原图像7-2a阈值化后的结果。经过阈值化后, 我们还想检验两类像素的分布是不是和所假设的分布一致。图7-4e和图7-4f 分别表示在高斯假设下物体像素和背景像素的直方图。根据这两个直方图, 我们能设计出一种迭代算法来估计每类新修正的参数, 然后利用这些参数来定义新的阈值。但是, 我们不能肯定这种迭代算法一定是收敛。还有, 这个结果比通过用滞后阈值法得的结果要差。这表明位置相邻标准和特征相似性的结合作用很大。

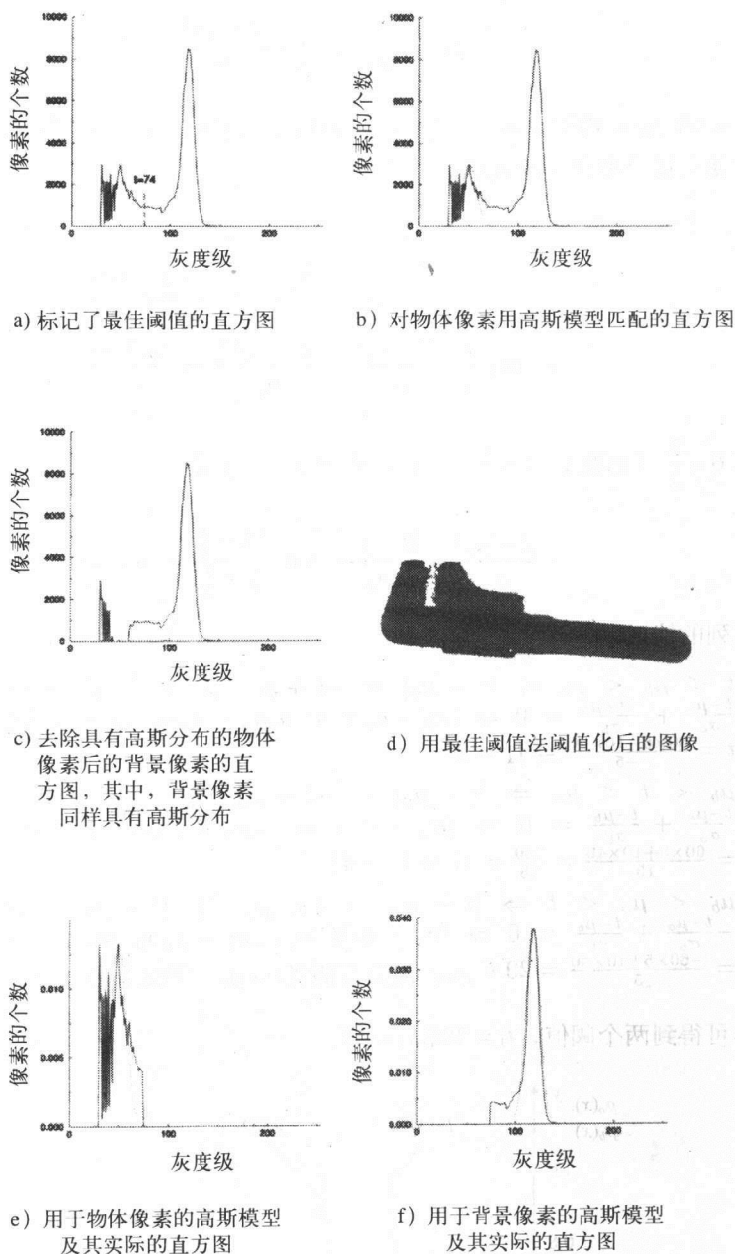


图7-4 最佳阈值法($t = 74$)

例7.6 图像中物体像素的灰度值的分布由下列的概率密度函数给出:

$$p_o = \frac{1}{2\sigma_o} \exp\left(-\frac{|x - \mu_o|}{\sigma_o}\right)$$

背景像素的灰度值的分布根据下列的概率密度函数:

$$p_b = \frac{1}{2\sigma_b} \exp\left(-\frac{|x - \mu_b|}{\sigma_b}\right)$$

如果 $\mu_o = 60$, $\mu_b = 40$ 和 $\sigma_o = 10$, $\sigma_b = 5$, 并且已知物体的像素占整幅图像的三分之二的条件下, 试确定使得错分率最小的阈值。

代入式(7-3)

$$\begin{aligned} \frac{\theta}{2\sigma_o} e^{-\frac{|t-\mu_o|}{\sigma_o}} &= \frac{1-\theta}{2\sigma_b} e^{-\frac{|t-\mu_b|}{\sigma_b}} \\ \Rightarrow \exp\left(-\frac{|t-\mu_o|}{\sigma_o} + \frac{|t-\mu_b|}{\sigma_b}\right) &= \frac{\sigma_o(1-\theta)}{\sigma_b\theta} \end{aligned}$$

我们有 $\theta = \frac{2}{3}$ (因此 $1-\theta = \frac{1}{3}$), $\sigma_o = 10$, $\sigma_b = 5$, 于是

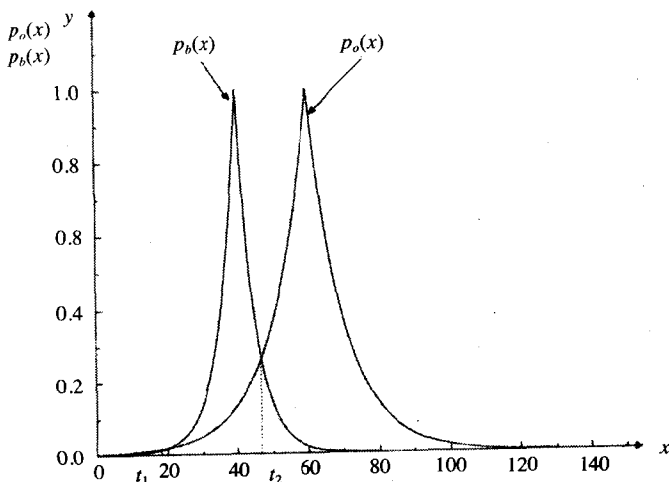
$$-\frac{|t-\mu_o|}{\sigma_o} + \frac{|t-\mu_b|}{\sigma_b} = \ln \frac{10 \times \frac{1}{3}}{5 \times \frac{2}{3}} = \ln 1 = 0$$

可得下列的情形:

- $t < \mu_b < \mu_o \Rightarrow |t - \mu_o| = -t + \mu_o$ 且 $|t - \mu_b| = -t + \mu_b \Rightarrow \frac{t-\mu_o}{\sigma_o} + \frac{-t+\mu_b}{\sigma_b} = 0 \Rightarrow (\sigma_b - \sigma_o)t = \mu_o\sigma_b - \sigma_o\mu_b \Rightarrow t = \frac{\mu_o\sigma_b - \sigma_o\mu_b}{\sigma_b - \sigma_o} \Rightarrow t = \frac{60 \times 5 - 10 \times 40}{-5} \Rightarrow t_1 = 20$
- $\mu_b < t < \mu_o \Rightarrow |t - \mu_o| = -t + \mu_o$ 且 $|t - \mu_b| = t - \mu_b \Rightarrow \frac{t-\mu_o}{\sigma_o} + \frac{t-\mu_b}{\sigma_b} = 0 \Rightarrow (\sigma_o + \sigma_b)t = \mu_o\sigma_b + \sigma_o\mu_b \Rightarrow t = \frac{\mu_o\sigma_b + \sigma_o\mu_b}{\sigma_b + \sigma_o} = \frac{60 \times 5 + 10 \times 40}{15} = \frac{700}{15} = 47 \Rightarrow t_2 = 47$
- $\mu_b < \mu_o < t \Rightarrow |t - \mu_o| = t - \mu_o$ 且 $|t - \mu_b| = t - \mu_b \Rightarrow -\frac{t-\mu_o}{\sigma_o} + \frac{t-\mu_b}{\sigma_b} = 0 \Rightarrow (\sigma_o - \sigma_b)t = \sigma_o\mu_b - \mu_o\sigma_b \Rightarrow t = \frac{-\mu_o\sigma_b + \sigma_o\mu_b}{\sigma_o - \sigma_b} = \frac{-60 \times 5 + 10 \times 40}{5} = 20 < \mu_o$, 因为有假设 $t > \mu_o$, 所以舍去

277

因此, 可得到两个阈值, $t_1 = 20$ 和 $t_2 = 47$ 。



只有把灰度值处于 t_1 和 t_2 中间的像素作为背景的像素才能使误差最小。

7.10 最小误差阈值方法的缺点是什么

这种方法有些缺点, 首先, 我们必须知道属于物体或属于背景的像素的先验概率, 其次, 必须知道两类像素的分布。通常, 将这两类像素的分布近似为正态分布, 但是即使这样, 我们也必须估计每个分布函数的 μ 和 σ 。

7.11 是否有一种可以不依赖于物体和背景像素分布模型的方法

Ostu提出了一种不依赖于物体和背景像素的概率密度分布模型的方法, 与前面的分析不同, 这种方法是直接在离散域得到的。

考虑一幅有 L 个灰度级的图像及其标准化后的直方图, 对每个灰度值 x , p_x 表示 x 出现的频数。假设我们将阈值设在 t 。如果我们现在要处理在深色背景下有一个浅色物体的图像, 那么分类为背景像素的比例为:

[278]

$$\theta(t) = \sum_{x=1}^t p_x \quad (7-8)$$

分类为物体像素的比例为:

$$1 - \theta(t) = \sum_{x=t+1}^L p_x \quad (7-9)$$

背景和物体像素的平均灰度值分别为:

$$\mu_b = \frac{\sum_{x=1}^t x p_x}{\sum_{x=1}^t p_x} \equiv \frac{\mu(t)}{\theta(t)} \quad (7-10)$$

$$\mu_o = \frac{\sum_{x=t+1}^L x p_x}{\sum_{x=t+1}^L p_x} = \frac{\sum_{x=1}^L x p_x - \sum_{x=1}^t x p_x}{1 - \theta(t)} = \frac{\mu - \mu(t)}{1 - \theta(t)} \quad (7-11)$$

这里我们定义 $\mu(t) \equiv \sum_{x=1}^t x p_x$, μ 是整幅图像的平均灰度值, 定义为:

$$\mu \equiv \frac{\sum_{x=1}^L x p_x}{\sum_{x=1}^L p_x} \quad (7-12)$$

同样, 可以定义由选定的阈值 t 确定的两个总体的方差:

$$\begin{aligned} \sigma_b^2 &\equiv \frac{\sum_{x=1}^t (x - \mu_b)^2 p_x}{\sum_{x=1}^t p_x} = \frac{1}{\theta(t)} \sum_{x=1}^t (x - \mu_b)^2 p_x \\ \sigma_o^2 &\equiv \frac{\sum_{x=t+1}^L (x - \mu_o)^2 p_x}{\sum_{x=t+1}^L p_x} = \frac{1}{1 - \theta(t)} \sum_{x=t+1}^L (x - \mu_o)^2 p_x \end{aligned} \quad (7-13)$$

下面考虑整幅图像所有像素分布的总体方差:

$$\sigma_T^2 = \sum_{x=1}^L (x - \mu)^2 p_x$$

将这个和式分解为两部分:

$$\sigma_T^2 = \sum_{x=1}^t (x - \mu)^2 p_x + \sum_{x=t+1}^L (x - \mu)^2 p_x$$

279

因为最后我们将从统计上定义这两个总体,所以在每个和式中加上并减去相应的均值:

$$\begin{aligned} \sigma_T^2 &= \sum_{x=1}^t (x - \mu_b + \mu_b - \mu)^2 p_x + \sum_{x=t+1}^L (x - \mu_o + \mu_o - \mu)^2 p_x \\ &= \sum_{x=1}^t (x - \mu_b)^2 p_x + \sum_{x=1}^t (\mu_b - \mu)^2 p_x + 2 \sum_{x=1}^t (x - \mu_b)(\mu_b - \mu) p_x \\ &\quad + \sum_{x=t+1}^L (x - \mu_o)^2 p_x + \sum_{x=t+1}^L (\mu_o - \mu)^2 p_x + 2 \sum_{x=t+1}^L (x - \mu_o)(\mu_o - \mu) p_x \end{aligned}$$

接着,利用式(7-13),将 σ_b^2 和 σ_o^2 代入每行左边的两个和式,注意到每行中间的两个和式可以通过式(7-8)和式(7-9)表示出来,因为 μ , μ_b 和 μ_o 是常数,不依赖对变量 x 的求和:

$$\begin{aligned} \sigma_T^2 &= \theta(t)\sigma_b^2 + (\mu_b - \mu)^2\theta(t) + 2(\mu_b - \mu) \sum_{x=1}^t (x - \mu_b)p_x \\ &\quad + (1 - \theta(t))\sigma_o^2 + (\mu_o - \mu)^2\theta(t) + 2(\mu_o - \mu) \sum_{x=t+1}^L (x - \mu_o)p_x \end{aligned}$$

这两项的和为零,例如:

$$\sum_{x=1}^t (x - \mu_b)p_x = \sum_{x=1}^t x p_x - \sum_{x=1}^t \mu_b p_x = \mu_b \theta(t) - \mu_b \theta(t) = 0$$

重新安排剩余各项的顺序:

$$\sigma_T^2 = \underbrace{\theta(t)\sigma_b^2 + (1 - \theta(t))\sigma_o^2}_{\text{依赖于类内方差的项}} + \underbrace{(\mu_b - \mu)^2\theta(t) + (\mu_o - \mu)^2(1 - \theta(t))}_{\text{依赖于类间方差的项}}$$

$$\equiv \sigma_W^2(t) + \sigma_B^2(t)$$

(7-14)

其中, $\sigma_W^2(t)$ 定义为类内方差, $\sigma_B^2(t)$ 定义为类间方差,显然, $\sigma_T^2(t)$ 是一个常数。我们希望确定 t 使 $\sigma_W^2(t)$ 尽可能小,也就是所创建的类尽可能密集,同时使 $\sigma_B^2(t)$ 尽可能的大。假设用 $\sigma_B^2(t)$ 来进行选择,也就是选择 t 使 $\sigma_B^2(t)$ 最大。我们将 μ_b 和 μ_o 代入 $\sigma_B^2(t)$ 的定义中,其中 μ_b 和 μ_o 通过式(7-10)和式(7-11)分别给出,可得:

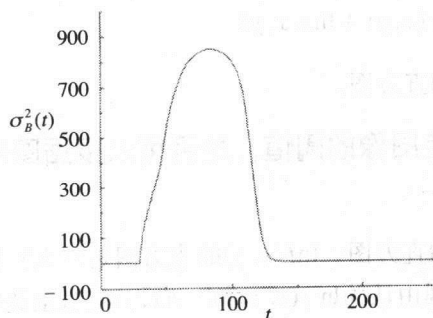
$$\begin{aligned}
 \sigma_B^2(t) &= (\mu_b - \mu)^2 \theta(t) + (\mu_o - \mu)^2 (1 - \theta(t)) \\
 &= \left[\frac{\mu(t)}{\theta(t)} - \mu \right]^2 \theta(t) + \left[\frac{\mu - \mu(t)}{1 - \theta(t)} - \mu \right]^2 (1 - \theta(t)) \\
 &= \frac{[\mu(t) - \mu\theta(t)]^2}{\theta(t)} + \frac{[\mu - \mu(t) - \mu + \mu\theta(t)]^2}{1 - \theta(t)} \\
 &= \frac{[\mu(t) - \mu\theta(t)]^2 [1 - \theta(t)] + \theta(t) [-\mu(t) + \mu\theta(t)]^2}{\theta(t)[1 - \theta(t)]} \\
 &= \frac{[\mu(t) - \mu\theta(t)]^2}{\theta(t)[1 - \theta(t)]} \quad (7-15)
 \end{aligned}$$

这个函数表示图像平均灰度值为 μ 的类间方差 $\sigma_B^2(t)$ ，只要知道阈值 t 所对应的直方图的值，就可以计算出 $\sigma_B^2(t)$ 值。

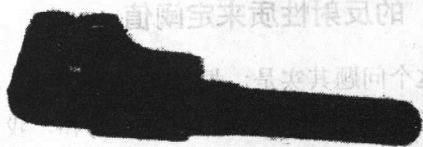
这个思想是：从直方图左端开始，通过计算 $\mu(t) = \sum_{x=1}^t xp_x$ 和 $\theta(t) = \sum_{x=1}^t p_x$ 的值，并将这些值代入式(7-15)，来检验每个灰度值作为使 $\sigma_B^2(t)$ 最大的阈值 t 的可能性。直到 $\sigma_B^2(t)$ 开始减少，我们才停止检验。这种方法可以确定阈值 t 使得 $\sigma_B^2(t)$ 最大，但是该方法的假设前提是函数 $\sigma_B^2(t)$ 性质好，且只有惟一的极大值。

例7.7 计算图7-2a中图像的Otsu阈值，并阈值化该图像。

图7-5a表示当 t 在所有的灰度值范围内变化时， $\sigma_B^2(t)$ 是如何变化的。当 $t=84$ 时，函数 $\sigma_B^2(t)$ 有第一个极大值，我们选择这个 t 来阈值化这幅图像，图7-5b表示其结果。注意到，这个结果与经验阈值方法结果（图7-2c）相比没有十分明显的差别，但比最佳阈值化方法的结果（图7-4d）要差，比通过滞后阈值法获得的结果更差。这更加说明了将空间和灰度特性相结合来阈值化图像的威力。



a) 对应 t 的 $\sigma_B(t)$



b) 利用Otsu法阈值化的图像

图7-5 Otsu阈值 ($t=84$)

7.12 Otsu方法有缺点吗

是的, 有一些缺点:

1) 尽管这个方法没有对概率密度函数 $p_o(x)$ 和 $p_b(x)$ 做任何假设, 仅仅利用均值和方差表达了这两个概率密度函数, 因此, 它假设了利用这两个统计量足以表示这两个概率密度函数, 但是事实上并非如此。

2) 当两个总体的分布十分不同时, 这个方法将会失效。当这两个总体大小相差很大时, $\sigma_b^2(t)$ 可能有两个极大值。实际上, 正确的极大值不必是全局的最大值, 这就是为什么实际操作时, 通过检验所选定阈值对应的直方图值 p_t , 从 $\sigma_b^2(t)$ 的全部极大值中确定的正确极大值实际上是一个谷 (即: $p_t < p_{\mu_o}, p_t < p_{\mu_b}$), 并且只有满足这个条件时, t 才能作为最佳阈值。

3) 上面提出的方法以直方图有双峰为前提条件, 即, 这幅图像有两个类。当图像中的类多于两个时, 这种方法必须修改, 以确定多个阈值。这种方法使得类间方差最大, 而类内方差最小。

4) 这种方法将图像分为了两类, 即使这种分法没有任何实际意义。事实上, 当一幅图像在变化的光照条件下, 该方法不能被直接应用。

282

7.13 如何对在变化光照下所获得的图像定阈值

在第4章“图像增强”中, 我们知道一幅图像从本质上说就是反射函数 $r(x, y)$ 和光照函数 $i(x, y)$ 的乘积, 其中反射函数 $r(x, y)$ 由物体可见表面的介质决定:

$$f(x, y) = r(x, y)i(x, y)$$

这样, 任何光照的空间变化都会引起对成像时所记录下的反射函数乘性的干扰, 我们可以将乘性干扰转变为加性干扰, 假如对图像取对数:

$$\ln f(x, y) = \ln r(x, y) + \ln i(x, y) \quad (7-16)$$

于是, 可以用 $\ln f(x, y)$ 的直方图代替 $f(x, y)$ 的直方图。

7.14 如果可以根据 $\ln f(x, y)$ 的直方图来定图像的阈值, 是否可以根据图像表面的反射性质来定阈值

这个问题其实是: 根据 $\ln r(x, y)$ 和 $\ln i(x, y)$ 的直方图, $\ln f(x, y)$ 的直方图是什么? 例如, 假如 $\ln f(x, y)$ 是 $\ln r(x, y)$ 和 $\ln i(x, y)$ 的和, 我们有理由认为 $\ln r(x, y)$ 和 $\ln i(x, y)$ 是从重迭中分离出来的函数, 于是通过阈值化 $\ln f(x, y)$, 就可以确定图像中的成分 $\ln r(x, y)$, 也就是我们感兴趣的成分。

定义新的变量:

$$z(x, y) \equiv \ln f(x, y)$$

$$\tilde{r}(x, y) \equiv \ln r(x, y)$$

$$\tilde{i}(x, y) \equiv \ln i(x, y)$$

因此, 式(7-16)可以写成:

$$z(x, y) = \tilde{r}(x, y) + \tilde{i}(x, y) \quad (7-17)$$

假如 $f(x, y)$ 、 $r(x, y)$ 和 $i(x, y)$ 是随机变量, 那么 $z(x, y)$ 、 $\tilde{r}(x, y)$ 和 $\tilde{i}(x, y)$ 也是随机变量, 因此这个问题可以改述为: 已知两个变量的直方图, 这两个随机变量和的直方图是什么? 直方图可以看成是一个概率密度函数, 所以这个问题可以改述为: 已知两个变量的概率密度函数, 两个随机变量和的概率密度函数是什么? 我们注意到一个随机变量的分布函数的微分就是这个随机变量的概率密度函数, 所以, 这个问题又可以改述为: 已知两个随机变量的分布函数或概率密度函数, 这两个随机变量的和的分布函数是什么? 在 (\tilde{i}, \tilde{r}) 空间中, 式(7-17)表示由给定的 z 值确定的一条直线。由定义可知:

283

z 的分布函数 $= P_z(u) = z \leq u$ 的概率 $= P(z \leq u)$

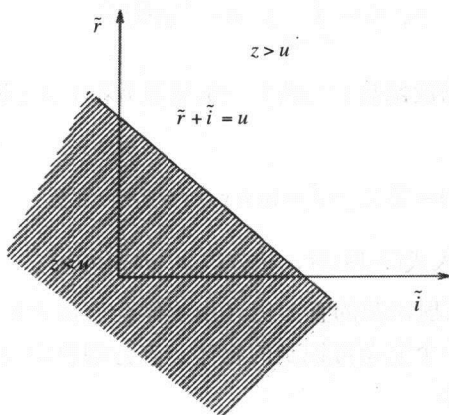


图7-6 在阴影半平面中, z 小于 u

直线 $\tilde{r} + \tilde{i} = u$ 将平面 (\tilde{i}, \tilde{r}) 分成两个半平面。一半 $z > u$, 另一半 $z < u$ 。 $z < u$ 的概率等于 (\tilde{i}, \tilde{r}) 的概率密度函数在整个 $z < u$ 半平面的积分 (见图7-6):

$$P_z(u) = \int_{\tilde{i}=-\infty}^{+\infty} \int_{\tilde{r}=-\infty}^{u-\tilde{i}} p_{\tilde{r}\tilde{i}}(\tilde{r}, \tilde{i}) d\tilde{r} d\tilde{i}$$

其中, $p_{\tilde{r}\tilde{i}}(\tilde{r}, \tilde{i})$ 表示随机变量 \tilde{r} 和 \tilde{i} 的联合概率密度函数。

为了求变量 z 的概率密度函数, 我们对 $P_z(u)$ 关于 u 求偏微分, 两次应用莱布尼兹法则 (Leibnitz's rule) (见方框B7.1), 一次应用莱布尼兹法则, 有:

284

$$\begin{aligned} f(x; \lambda) &\rightarrow \int_{\tilde{r}=-\infty}^{u-\tilde{i}} p_{\tilde{r}\tilde{i}}(\tilde{r}, \tilde{i}) d\tilde{r} \\ b(\lambda) &\rightarrow +\infty \\ a(\lambda) &\rightarrow -\infty \end{aligned}$$

当需要对 $f(x; \lambda)$ 关于 u 求微分时, 再次应用莱布尼兹法则, 其中 $f(x; \lambda)$ 本身是依赖于所要微

284

分的参数 u 的积分:

$$\begin{aligned} p_z(u) &= \frac{dP_z u}{du} \int_{\tilde{i}=-\infty}^{\infty} \frac{d}{du} \left[\int_{\tilde{r}=-\infty}^{u-\tilde{i}} p_{\tilde{r}\tilde{i}}(\tilde{r}, \tilde{i}) d\tilde{r} \right] d\tilde{i} \\ &= \int_{\tilde{i}=-\infty}^{\infty} p_{\tilde{r}\tilde{i}}(u - \tilde{i}, \tilde{i}) d\tilde{i} \end{aligned} \quad (7-18)$$

这两个随机变量 \tilde{r} 和 \tilde{i} 是相互独立的, 其中 \tilde{r} 和图像表面有关, 而 \tilde{i} 和光源有关, 因此它们的联合概率密度函数可以写成它们的概率密度函数的乘积:

$$p_{\tilde{r}\tilde{i}}(\tilde{r}, \tilde{i}) = p_{\tilde{r}}(\tilde{r})p_{\tilde{i}}(\tilde{i})$$

将上式代入式(7-18), 得:

$$p_z(u) = \int_{-\infty}^{\infty} p_{\tilde{r}}(u - \tilde{i})p_{\tilde{i}}(\tilde{i}) d\tilde{i} \quad (7-19)$$

这表明 z 的直方图 (= 概率密度函数) 与两个随机变量 \tilde{r} 和 \tilde{i} 直方图的卷积相等。

假如光照是均衡的, 于是:

$$i(x, y) = \text{常数} \Rightarrow \tilde{i} = \ln i(x, y) = \tilde{i}_0 = \text{常数}$$

于是, $p_{\tilde{i}}(\tilde{i}) = \delta(\tilde{i} - \tilde{i}_0)$, 代入式(7-19)后, 积分得 $p_z(u) = p_{\tilde{r}}(u)$ 。

这说明, 在均衡光照下, 反射函数的直方图 (由物体介质确定) 基本上不受影响, 但是如果光照不是均衡的, 即使有一个完全可识别的物体, 直方图也会变形的非常严重, 在这种情况下, 各种阈值化方法会失效。

7.15 如果直接阈值化方法在变化光照的情况下失效, 应该如何处理

有两种方法可以解决在变化光照下出现的问题:

1) 将图像分割成一些小块, 每一块的光照大致均衡, 将每块单独看做一幅图像, 求出其直方图, 并阈值化。当我们合并相邻块时, 如果相邻块的阈值发生了跳跃, 我们要进行相应的调整。

2) 利用具有均衡表面反射率的图像, 并且用 $i(x, y)$ 除这幅图像 $f(x, y)$, 构造一幅只与光照场有关的图像。它等价于从 $z(x, y)$ 中减去光照成分 $\tilde{i}(x, y)$, 于是 $\frac{f(x, y)}{i(x, y)}$ 带有一个干扰值, 例

如, $i(0, 0)$, $\frac{f(x, y)}{i(x, y)}$ 将会使整幅图像具有相同光照条件, 在修正的图像中进行阈值化处理。

• 例7-8 阈值化图7-7a中的图像。

这幅图像从左到右表现出了光照的变化, 图7-7b显示其直方图, 利用Otsu方法, 我们可以确定阈值 $t = 75$, 图7-7c表示用该阈值对图像阈值化后的结果, 图7-7d表示了从左到右将图像分成四个子图像, 然后将Otsu方法分别应用于每个子图像后所得的结果。

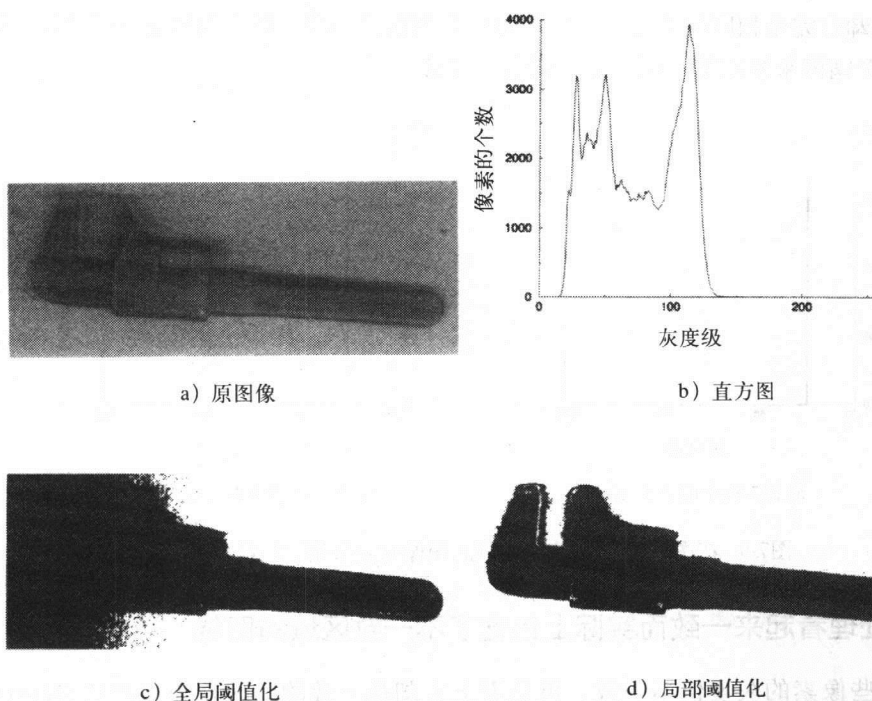


图7-7 光照变化条件下，图像全局阈值化和局部阈值化对比

286

7.16 阈值方法有哪些缺点

在图像分割中，除了滞后阈值法利用了像素的空间邻接关系之外，其他的阈值法根本没有考虑像素的空间邻接关系，而只是利用了像素的灰度值。

例如，考虑下面图7-8中的两幅图像。

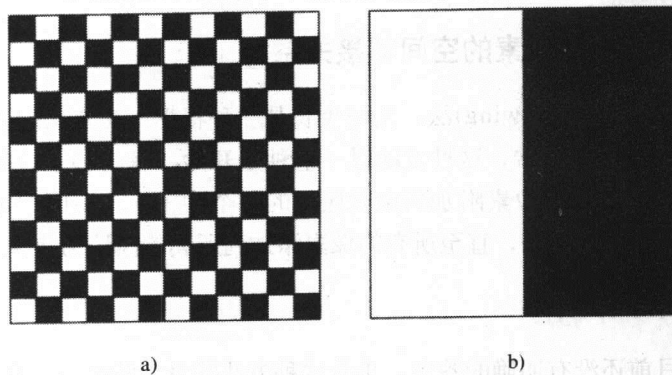


图7-8 两幅有相同的直方图却完全不同的图像

显然，第一幅图像有一致的区域，而第二幅图像包含了两个完全不同的区域。尽管这样，

288

图7-9表示它们却有完全相同的直方图，它们的直方图有两个峰，我们很容易选择阈值。但是，如果我们用这个阈值来分割第一幅图像，则毫无意义。

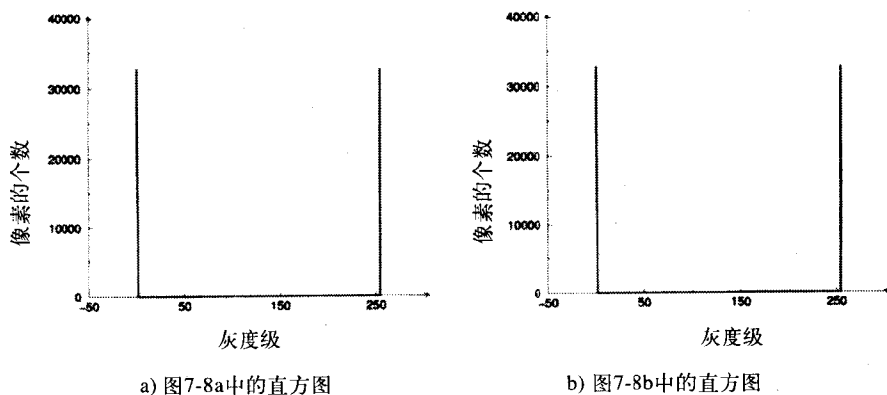


图7-9 在图7-8中两幅完全不同的图像有完全相同的直方图

287

7.17 如何处理看起来一致而实际上包含了不一致区域的图像

我们把那些像素的灰度值不一致，但是看上去却是一致的区域称为纹理区域(textured region)。从分割的目的出发，每个像素不能仅仅通过它的灰度值来刻画，还要通过其周围一小块区域内像素灰度值的变化来度量。由纹理图像分割所引出的问题可以采用多种属性刻画图像来解决。我们设想每个像素不是由一个数字而是通过多个数字组成的向量来刻画的，向量中的每个分量度量了在像素位置的某个方面，于是，每个像素通过多维空间中的一个点来表示。在这个多维空间中，沿每个轴度量了一个数字，一个特征(feature)。属于同一个区域的像素有相似的或者相同的属性值，于是，它们可以构成一个聚类。这样一来，问题就转变为在一个多维空间中确定像素所属的聚类了。现在我们需要处理多维直方图，它本质上同一维直方图方法相似。我们可以采用几种聚类(clustering)的方法，但是聚类方法属于模式识别(Pattern Recognition)领域探讨的问题，它已经不属于本书所讨论的范畴。

7.18 有哪些方法考虑了像素的空间邻接关系

有一种区域生长(region growing)法。区域生长是一种根据事先定义的准则将像素或子区域聚合成更大区域的过程。通常，这种方法以一组种子开始，将与种子性质相似（诸如灰度级或颜色的特定范畴）的相邻像素附加到生长区域的每个种子上，所以，每个种子通过积累越来越多的相邻像素慢慢地生长，直至所有的像素添加到所属的区域为止。

7.19 如何选择种子像素

对于这个问题目前还没有明确的答案，也是这种方法的最大的缺点。在一些实际应用中，种子像素的选择比较容易，例如，在红外线的目标跟踪中，目标看上去比较明亮，我们可以选择那些最亮的像素作为种子。有一种不需要事先决定生长区域或种子数的方法就是拆分和合成。

288

7.20 拆分和合成方法是如何进行的

首先, 将整幅图像看做一个区域, 如果在这个区域内的属性值范围超过了预先定义的值, 就将区域分成四个区域, 每个区域采用上述同样的方法进行检验, 直到按这种方法创建的正方形区域所包含的像素属性值的范围小于预先定义的值, 最后, 将具有相同属性的相邻区域进行合并。

图7-10展示了一个例子, 为了简单起见, 考虑一幅 8×8 的二值图像, 下面的树结构表示了图像不断地拆分为四个区域的过程, 这种树称为四分树 (quadtree)。

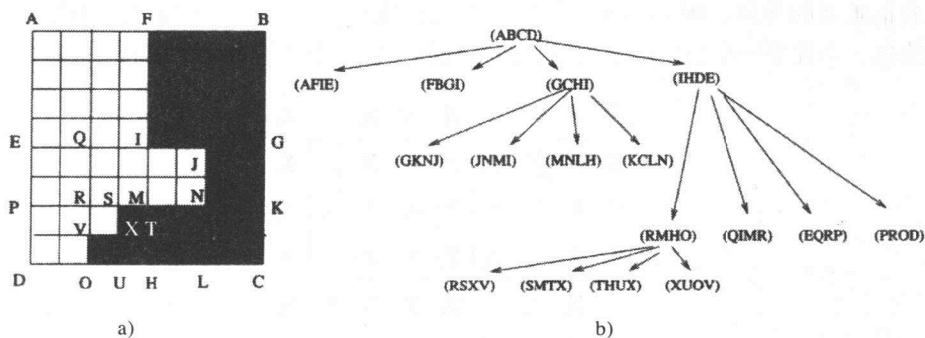


图7-10 通过拆分来分割图像

最后, 我们可以得到下面的区域:

(AFIE) (FBGI) (GKNJ) (JNMI)(MNLH)(KCLN)
(RSXV)(SMTX)(THUX)(XUOV)(QIMR)(EQR)(PROD)

即, 所有四分树的孩子。为了合并, 检验任何两个相邻的区域, 最后, 只有两个主要的不规则区域合并。当图像是每边包含 $N = 2^n$ 个像素的正方形时, 上述的四分树结构是非常常用的。

拆分和合并方法通常从四分树的某一个中间级开始 (即, 从一些大小为 $2^l \times 2^l$ 的块开始, 其中, $l < n$), 为了进一步将块拆分成四个正方形子块, 要检验每个块, 同时, 为了合并, 需要检验任何相邻的块, 最后, 为了合并相邻的块, 再次检验任何相邻的块。

7.21 与考虑像素之间的相似性相反, 是否可以考虑区域之间的不相似性来进行图像分割

可以, 在这种方法中, 我们检验相邻像素的差异, 属于不同区域的像素有不同的属性值, 因此, 我们假定有一个边界可以分开它们, 这个边界称为边缘, 这种方法称为边缘检测。

7.22 如何度量相邻像素之间的不相似性

在图像的每个位置上滑动一个窗口, 计算包含窗口左半部分和右半部分的像素统计量, 对比两者, 可以认为那些统计量变化最大的地方就是边缘区域。

例如, 针对图7-11中的 8×8 图像, 其中每个X表示一个像素。图像中出现的 3×7 的矩形便是滑动的窗口, 除去那些靠近图像边缘的像素, 窗口的中心与图像的像素逐个重合。于是, 计算分别落在窗口左边(A部分)和窗口右边(B部分)的9个像素点的统计量, 然后把它们之间的差值记录在像素点O上。例如, 分别计算图像落在窗口左右两部分上的像素点的灰度值的标准差, 记为 σ_A 和 σ_B ; 计算整个窗口里面的像素点的灰度值的标准差, 记为 σ ; 将 $E = 2\sigma - \sigma_A - \sigma_B$ 记录在中心像素O上。接着, 水平滑动这个窗口, 遍历整幅图像, 得到的局部极大值可以作为待选的垂直边缘点, 当E值大于某个阈值时便可以认为局部极大值是此邻接区域的垂直边缘点。可以将上述窗口旋转 90° , 垂直滑动窗口再一次遍历整幅图像。很明显, 这里窗口的大小起着很重要的作用, 因为为了计算窗口的统计性质需要足够大的窗口, 而为了刚好使一个区域的每一半置于一个窗口中, 避免邻接区域的干扰, 需要尽量小的窗口。

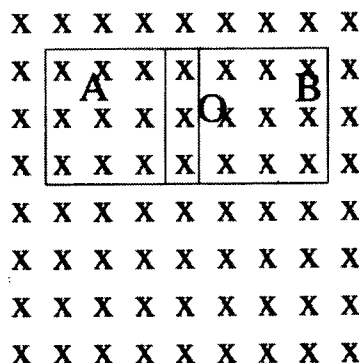


图7-11 使用滑动窗口方法度量图像两个区域的不相似性

7.23 可以选择的最小窗口是什么样的

可以选择的最小窗口是只包含两个相邻像素的窗口。该窗口中惟一能计算的“统计”量是这两个像素点的灰度的差。当这个差足够大, 我们便认为这两个像素产生了一个边缘, 当然, 仅仅是两个像素点灰度之差不能够说是统计量, 而只是图像密度函数的一阶导数的估算量, 其中取差分的方向就是空域变化的方向。这是因为在离散情况下, 一阶导数可以用一阶差分近似表达:

$$\begin{aligned}\Delta f_x &= f(i+1, j) - f(i, j) \\ \Delta f_y &= f(i, j+1) - f(i, j)\end{aligned}$$

在每个像素点上计算 Δf_x 等价于沿x轴方向将图像与一个形式为 $\begin{bmatrix} -1 & +1 \end{bmatrix}$ 的模板 (滤波器)

进行卷积; 而计算 Δf_y 等价于沿y轴方向与滤波器 $\begin{bmatrix} -1 \\ +1 \end{bmatrix}$ 做卷积运算。

最简单的边缘检测方法就是让图像与这两个模板做卷积运算得到两个输出。注意到两个模板的长度为偶数, 当它们在图像中滑动时, 它们的中心并不与图像中任何特定的像素相关联。因此, 每个计算的输出都应该记录在两个相邻像素之间的位置上。我们称这些位置为图像的二重 (dual) 网格。实际应用上, 我们很少用到二重网格。我们经常采用一个约定并且

保持一致。比如，我们经常会把得到的输出记录在模板的第一像素上，当需要用到它的时候，我们只要记住，这个值实际上是度量两个相邻像素的差值，它的实际位置是在记录该值的像素的左边或下边半个像素距离的位置上。有了这样的约定，从现在开始，为了简单起见，我们就叫它边缘像素。

对于由图像与模板 $\begin{bmatrix} -1 & +1 \end{bmatrix}$ 卷积而产生的第一个输出，只要像素的绝对值比它的左邻像素和它的右邻像素都大，它就是一个待选的垂直边缘像素；对于由图像与模板 $\begin{bmatrix} -1 \\ +1 \end{bmatrix}$ 卷积而产生的第二个输出，只要像素绝对值比它的上邻像素和它的下邻像素都大，它就是一个待选的水平边缘像素。确定局部极大值作为待选边缘像素的过程称为非极大抑制 (non-maxima suppression)。

在没有噪声的情况下，这种方法很容易就将图像中灰度变化不连续的地方找出来。

7.24 当图像有噪声的时候会发生什么情况

在噪声存在的情况下，在度量图像中像素差值时，即使是很小的而且不相关的灰度波动都有可能被放大。通常情况下应该先用一个低通滤波器对图像进行滤波，然后再去找那些局部差值。图7-12显示一幅原始图像和它分别和 $\begin{bmatrix} -1 & +1 \end{bmatrix}$ 与 $\begin{bmatrix} -1 \\ +1 \end{bmatrix}$ 卷积滤波器在水平方向与垂直方向进行卷积的结果。将这两个卷积的结果先进行平方，然后相加，再做平方根，得到每个像素的梯度值。图7-13显示的是用最小卷积滤波器结果，以及考虑到图像中有噪声的情况下用一些更复杂的滤波器进行滤波的结果。

291

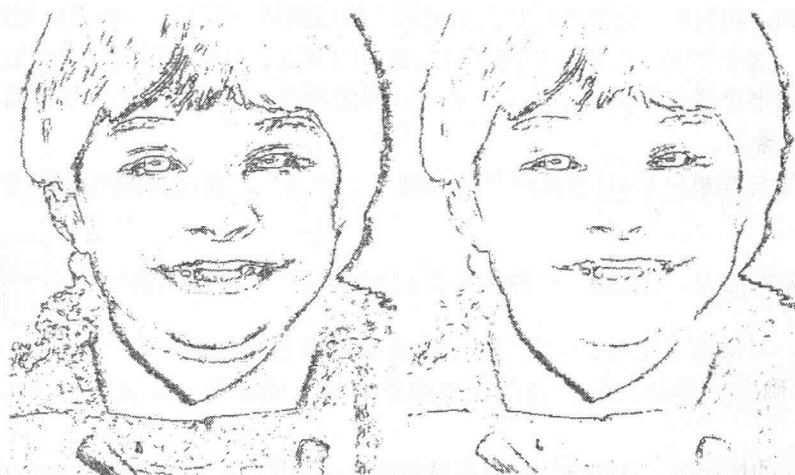


a) 原始图像

b) 不应用任何光滑预处理，仅通过简单的差分运算来计算梯度图像

图 7-12

为了显示的需要，将梯度图像应用直方图均衡化。



a) 在没有进行光滑处理而获得的梯度图中使用简单阈值法得到的结果

b) 使用更大的阈值得到的结果, 有些噪声被祛除了, 但同时有些有用的图像信息也被丢失了



c) 使用Sobel模板, 先在一个方向进行光滑, 再在垂直方向做差分运算得到的结果

d) 用尺寸为7×7的最优滤波器得到的结果

图7-13 在以上各种的情况下, 使用相同的阈值

下面考虑一个一维信号的例子。假如使用一个简单的平均化过程作为低通滤波器, 也就是说, 通过用三个连续像素的灰度平均值替换每一个灰度值来光滑信号:

$$A_i \equiv \frac{I_{i-1} + I_i + I_{i+1}}{3} \quad (7-20)$$

接着, 计算位置*i*的值和它的左邻与右邻间的差值, 再将两个值平均, 从而得到位置*i*的导数的一个估算:

$$F_i \equiv \frac{(A_{i+1} - A_i) + (A_i - A_{i-1}))}{2} = \frac{A_{i+1} - A_{i-1}}{2} \quad (7-21)$$

如果将式(7-20)代入式(7-21),可以得到:

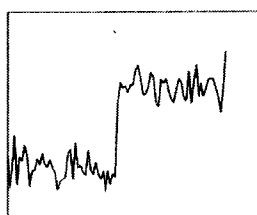
$$F_i = \frac{1}{6}[I_{i+2} + I_{i+1} - I_{i-1} - I_{i-2}] \quad (7-22)$$

从这个例子可以清楚地看到,只要使用足够大的模板,就可以把光滑及寻找差值两个线性运算合并为一个运算。在这种情况下,在每个位置做一阶差分等同于使用模板

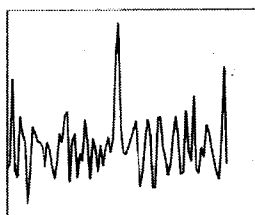
$-\frac{1}{6}$	$-\frac{1}{6}$	0	$\frac{1}{6}$	$\frac{1}{6}$
----------------	----------------	---	---------------	---------------

来做估算。很明显,模板长度越大,光滑效果越好。但是,模板长度越

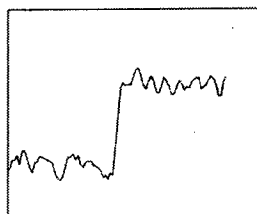
大,边缘就会变得越模糊,从而导致它的位置不容易精确获取(见图7-14)。



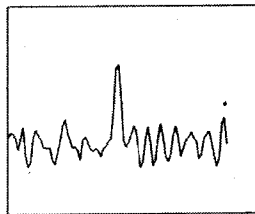
a) 一个有噪声的信号



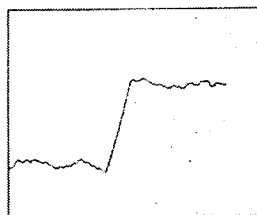
b) 用差分法对图7-14a求一阶导数的结果



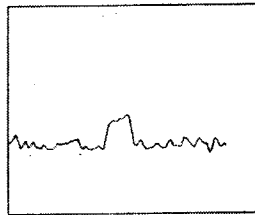
c) 每三个采样点做一次平均来进行光滑的结果



d) 用差分法对图7-14c求一阶导数的结果



e) 每十个采样点做一次平均来进行光滑的结果



f) 用差分法对图7-14e求一阶导数的结果

图7-14 左列显示了一个被噪声污染的图像和对它进行光滑的两种结果,右边是通过简单差分运算估算一阶导数的结果。边缘区域显示为一个尖锐的峰,但是,噪声残留在信号里面越多,就会出现越多的次要峰,而图像越光滑,主峰就变得越钝

对于二维图像, 必须使用二维模板。同时带有最少光滑运算和差分运算的最小的模板就是 3×3 模板。在这种情况下, 可以选择在一个方向上做光滑, 而在另一个方向上做差分运算, 这就意味着, 这个二维模板是先应用 3×1 光滑模板再应用 1×3 差分模板层叠得到的, 反之亦然。总而言之, 这个二维 3×3 模板具有形式:

a_{11}	a_{12}	a_{13}
a_{21}	a_{22}	a_{23}
a_{31}	a_{32}	a_{33}

7.25 对于边缘检测如何选择 3×3 模板的权重

使用一个模板来计算 Δf_x , 同时使用另外一个模板来计算 Δf_y 。这些模板必须遵循以下原则:

1) 计算 Δf_x 的模板必须是由计算 Δf_y 的模板旋转 90° 而得来的。所以以后只需考虑计算 Δf_y 的模板, 计算得到的值会记录在中心像素点上。

2) 由于不希望中心像素的左邻像素或右邻像素有不同的权重, 因此模板左右两列必须有同样的权重, 于是, 这个 3×3 模板就必须有以下的形式:

a_{11}	a_{12}	a_{11}
a_{21}	a_{22}	a_{21}
a_{31}	a_{32}	a_{31}

3) 为了找到局部差值, 可以从中心像素点“后面”的像素值减去中心像素点“前面”的像素值, 同时, 还要让两个被减的像素应有相同的权重, 因此, 该 3×3 模板就必须有以下的形式:

a_{11}	a_{12}	a_{11}
a_{21}	a_{22}	a_{21}
$-a_{11}$	$-a_{12}$	$-a_{11}$

4) 如果图像是完全光滑的, 希望得到的是零响应, 那么所有的权重的和必须是零, 于是, $a_{22} = -2a_{21}$:

a_{11}	a_{12}	a_{11}
a_{21}	$-2a_{21}$	a_{21}
$-a_{11}$	$-a_{12}$	$-a_{11}$

5) 当信号是光滑的时候, 在列的方向上做差分运算, 希望在每一列都是0输出。因此, $a_{21} = 0$:

a_{11}	a_{12}	a_{11}
0	0	0
$-a_{11}$	$-a_{12}$	$-a_{11}$

最后, 把所有的权重都除以 a_{11} , 于是整个模板只依赖于一个参数:

1	K	1
0	0	0
-1	$-K$	-1

(7-23)

7.26 参数 K 的最佳值是什么

可以看出, 图像边缘的方向跟图像中轴是对齐的, 所以它不会受到差分运算的影响。选择 $K=2$, 就可以得到两个方向上对图像求导的Sobel模板。

1	2	1	-1	0	1
0	0	0	-2	0	2
-1	-2	-1	-1	0	1

注意到第二个模板做了些改变, 即从中心像素“前面”的像素值中减去中心像素“后面”的像素值。这样做的目的是, 使用这两个模板得到的梯度向量的分量再计算图像边缘方向, 边缘的方向是沿水平轴逆时针方向测量的 (见方框B7.2)。

B7.2 Sobel模板的推导

边缘具有大小和方向, 即:

$$\text{大小: } E(i, j) = \sqrt{[\Delta f_x(i, j)]^2 + [\Delta f_y(i, j)]^2}$$

$$\text{方向: } a(i, j) = \tan^{-1} \frac{\Delta f_y(i, j)}{\Delta f_x(i, j)}$$

我们的目标是确定(7-23)所示模板中的参数 K , 从而使运算结果尽可能真实地反映非离散图像对应的 E 和 a 的实际值。

$$E = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

$$a = \tan^{-1} \frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}}$$

设想一下场景中有一段阶跃边缘通过某个像素点的中心, 假设每个像素点都是一个 1×1 的块, 边缘的倾斜角度为 θ 而且 θ 足够小, 使得边缘切割线段 AB 和 CD 而不是切割 AC 和 BD ($0 < \theta < \tan^{-1}(\frac{1}{3})$) (参见图7-15)。

另外, 假设边缘的左边是一个较暗的区域, 灰度值是 G_1 ; 边缘的右边是一个明亮的区域, 灰度值为 G_2 。很明显, 在模板里面的像素值是:

$$f(i-1, j-1) = f(i-1, j) = f(i-1, j+1) = G_1$$

$$f(i+1, j-1) = f(i+1, j) = f(i+1, j+1) = G_2$$

中间列的像素具有混合值。假设每个像素点像一个 1×1 的块，用括号内的多边形的名称来表示这个区域的面积，于是，像素ABIL的值是：

$$f(i, j-1) = G_1(AFJL) + G_2(FBIJ) = G_1\left[\frac{1}{2} - (FGHJ)\right] + G_2\left[\frac{1}{2} + (FGHJ)\right] \quad (7-24)$$

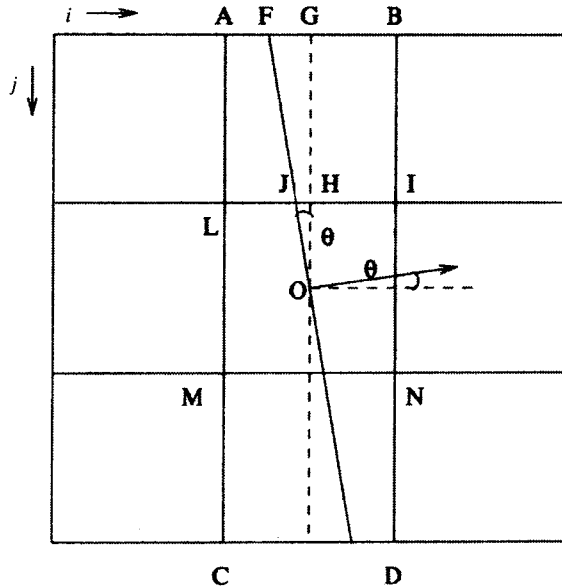


图7-15 放大图像在像素点 (i, j) 周围的一个 3×3 的片断

需要求梯形FGHJ的面积。由从三角形OJH和OFG可以得出：

$$JH = \frac{1}{2} \tan \theta, \quad FG = \frac{3}{2} \tan \theta$$

于是有 $(FGJH) = \frac{1}{2}(JH + FG) = \tan \theta$ ，将该式代入式(7.24)就可以得到：

$$f(i, j-1) = G_1\left(\frac{1}{2} - \tan \theta\right) + G_2\left(\frac{1}{2} + \tan \theta\right)$$

由对称性可知：

$$f(i, j+1) = G_2\left(\frac{1}{2} - \tan \theta\right) + G_1\left(\frac{1}{2} + \tan \theta\right)$$

显然有：

$$f(i, j) = \frac{G_1 + G_2}{2}$$

现在看看, 采用式(7-23)可以计算出来什么结果:

$$\begin{aligned}
 \Delta f_x &= f(i+1, j+1) + f(i+1, j-1) + Kf(i+1, j) \\
 &\quad - [f(i-1, j+1) + f(i-1, j-1) + Kf(i-1, j)] \\
 &= (G_2 + G_2 + KG_2) - (G_1 + G_1 + KG_1) = (G_2 - G_1)(2 + K) \\
 \Delta f_y &= -[f(i-1, j+1) + f(i+1, j+1) + Kf(i, j+1)] \\
 &\quad + f(i-1, j-1) + f(i+1, j-1) + Kf(i, j-1) \\
 &= -G_1 - G_2 - KG_2 \left(\frac{1}{2} - \tan \theta \right) - KG_1 \left(\frac{1}{2} + \tan \theta \right) + G_1 + G_2 \\
 &\quad + KG_1 \left(\frac{1}{2} - \tan \theta \right) + KG_2 \left(\frac{1}{2} + \tan \theta \right) \\
 &= -K(G_2 - G_1) \left(\frac{1}{2} - \tan \theta \right) + K(G_2 - G_1) \left(\frac{1}{2} + \tan \theta \right) \\
 &= K(G_2 - G_1) \left(-\frac{1}{2} + \tan \theta + \frac{1}{2} + \tan \theta \right) \\
 &= 2K(G_2 - G_1) \tan \theta
 \end{aligned}$$

边缘的大小是:

$$\begin{aligned}
 E &= \sqrt{(G_2 - G_1)^2(2 + K)^2 + (2K)^2(G_2 - G_1)^2 \tan^2 \theta} \\
 &= (G_2 - G_1)(2 + K) \sqrt{1 + \left(\frac{2K}{2 + K} \right)^2 \tan^2 \theta}
 \end{aligned}$$

而边缘的方向是:

$$\tan \alpha = \frac{\Delta f_y}{\Delta f_x} = \frac{2K \tan \theta}{2 + K}$$

注意到, 如果取 $K = 2$:

$$\tan \alpha = \tan \theta$$

即, 计算得到的边缘的方向和它实际的方向是一致的。

对于 $\tan^{-1}(\frac{1}{3}) < \theta < 45^\circ$ 的情形, 我们可以类似地计算。在那种情况下, 边缘的方向会

有些改变。

- 例7.9 将 $O(i, j)$ 记为输入值 $I(i, j)$ 的函数, 它的意义是图像在 (i, j) 位置上的沿 i 轴方向与Sobel模板做卷积运算的结果, 写下这个公式就是:

(注意: 这里忽略了边缘效应, 即, 假设 (i, j) 离图像的边界足够远。)

$$O(i, j) = -I(i-1, j-1) - 2I(i-1, j) - I(i-1, j+1) \\ + I(i+1, j-1) + 2I(i+1, j) + I(i+1, j+1)$$

例7.10 如果有一幅 3×3 的图像, 可以把它表示成一个 9×1 的向量, 我们希望构造一个 9×9 的矩阵, 当这个矩阵作用于图像向量时会产生另一个向量, 该向量的每个分量是图像沿着 i 方向的梯度分量的一个估算。(为了解决边界效应问题, 假设图像是在任何方向都周期重复的。)

299

考虑一幅 3×3 图像:

$$\begin{matrix} & \xrightarrow{i} \\ j \downarrow & \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix} \end{matrix}$$

周期性的重复就意味图像有这样一种形式:

$$\begin{matrix} f_{33} & f_{31} & f_{32} & f_{33} & f_{31} \\ f_{13} & \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix} & f_{11} \\ f_{23} & & f_{21} \\ f_{33} & & f_{31} \\ f_{13} & f_{11} & f_{12} & f_{13} & f_{11} \end{matrix}$$

应用例7.9的结果, 可以得到在位置 $(1, 1)$ 的一阶导数是:

$$f_{32} + 2f_{12} + f_{22} - f_{33} - 2f_{13} - f_{23}$$

图像的向量表示形式为:

$$\begin{pmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{pmatrix}$$

如果将一个 9×9 的矩阵作用于该图像, 可以注意到为了得到预定的输出, 矩阵的第一行必须是:

$$0 \quad 0 \quad 0 \quad 2 \quad 1 \quad 1 \quad -2 \quad -1 \quad -1$$

在位置(1, 2), 即在像素 f_{21} 上的一阶导数是:

$$f_{12} + 2f_{22} + f_{32} - f_{13} - 2f_{23} - f_{33}$$

于是, 9×9 矩阵的第二行必须是:

$$0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 1 \quad -1 \quad -2 \quad -1$$

在位置(1, 3)上的一阶导数是:

$$f_{22} + 2f_{32} + f_{12} - f_{23} - 2f_{33} - f_{13}$$

300

于是, 9×9 矩阵的第三行必须是:

$$0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 2 \quad -1 \quad -1 \quad -2$$

按这种方法, 可以得出矩阵的最后形式应该是:

$$\begin{pmatrix} 0 & 0 & 0 & 2 & 1 & 1 & -2 & -1 & -1 \\ 0 & 0 & 0 & 1 & 2 & 1 & -1 & -2 & -1 \\ 0 & 0 & 0 & 1 & 1 & 2 & -1 & -1 & -2 \\ -2 & -1 & -1 & 0 & 0 & 0 & 2 & 1 & 1 \\ -1 & -2 & -1 & 0 & 0 & 0 & 1 & 2 & 1 \\ -1 & -1 & -2 & 0 & 0 & 0 & 1 & 1 & 2 \\ 2 & 1 & 1 & -2 & -1 & -1 & 0 & 0 & 0 \\ 1 & 2 & 1 & -1 & -2 & -1 & 0 & 0 & 0 \\ 1 & 1 & 2 & -1 & -1 & -2 & 0 & 0 & 0 \end{pmatrix}$$

例7.11 使用从例7.10得到的矩阵, 对下列图像沿*i*轴方向计算一阶导数:

$$\begin{pmatrix} 3 & 1 & 0 \\ 3 & 1 & 0 \\ 3 & 1 & 0 \end{pmatrix}$$

那么图像中央沿着*i*轴方向的梯度分量是多少呢?

$$\begin{pmatrix} 0 & 0 & 0 & 2 & 1 & 1 & -2 & -1 & -1 \\ 0 & 0 & 0 & 1 & 2 & 1 & -1 & -2 & -1 \\ 0 & 0 & 0 & 1 & 1 & 2 & -1 & -1 & -2 \\ -2 & -1 & -1 & 0 & 0 & 0 & 2 & 1 & 1 \\ -1 & -2 & -1 & 0 & 0 & 0 & 1 & 2 & 1 \\ -1 & -1 & -2 & 0 & 0 & 0 & 1 & 1 & 2 \\ 2 & 1 & 1 & -2 & -1 & -1 & 0 & 0 & 0 \\ 1 & 2 & 1 & -1 & -2 & -1 & 0 & 0 & 0 \\ 1 & 1 & 2 & -1 & -1 & -2 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 3 \\ 3 \\ 3 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \\ 4 \\ -12 \\ -12 \\ -12 \\ 8 \\ 8 \\ 8 \end{pmatrix}$$

图像中央沿着*i*轴方向的梯度分量是-12。

301

7.27 一般情况下，如何决定一个像素是否是边缘像素

边缘就是图像中图像函数变化的点。由于图像是一个二维函数，所以，要找到这些点，必须计算图像函数的梯度 $\nabla f(x, y)$ 。该二维函数的梯度是一个二维向量，它的分量是图像函数沿着两个正交方向的偏导数。在离散的情况下，计算这些偏导数相当于使用诸如Sobel之类的模板分别计算两个正交方向上相邻像素的差值。如果用这些模板和图像做卷积运算，图像的每一个像素处就会产生一个梯度向量，边缘就是沿着梯度向量的方向（也就是像素值的梯度方向）的局部最大的地方。为了达到这个目的，必须对比下面两个梯度值的大小，一个是当前像素位置的梯度值，另一个是沿着梯度方向两边单位间隔位置上的梯度值。一般来说，这些值是不知道的，因为它们有可能位于像素之间。于是，要么用一块局部曲面去拟合这些梯度值，再来估算所要求的像素之间位置的梯度大小，要么就使用插值的方法在已知的整数位置进行插值来计算这些位置的梯度值。

在这个非极大抑制（non-maxima suppression）过程后，将保留下来的那些梯度向量的值进行阈值化。在边缘图（edge map）中，只有当梯度值大于阈值的那些像素点才被确定为边缘像素（见图7-16）。

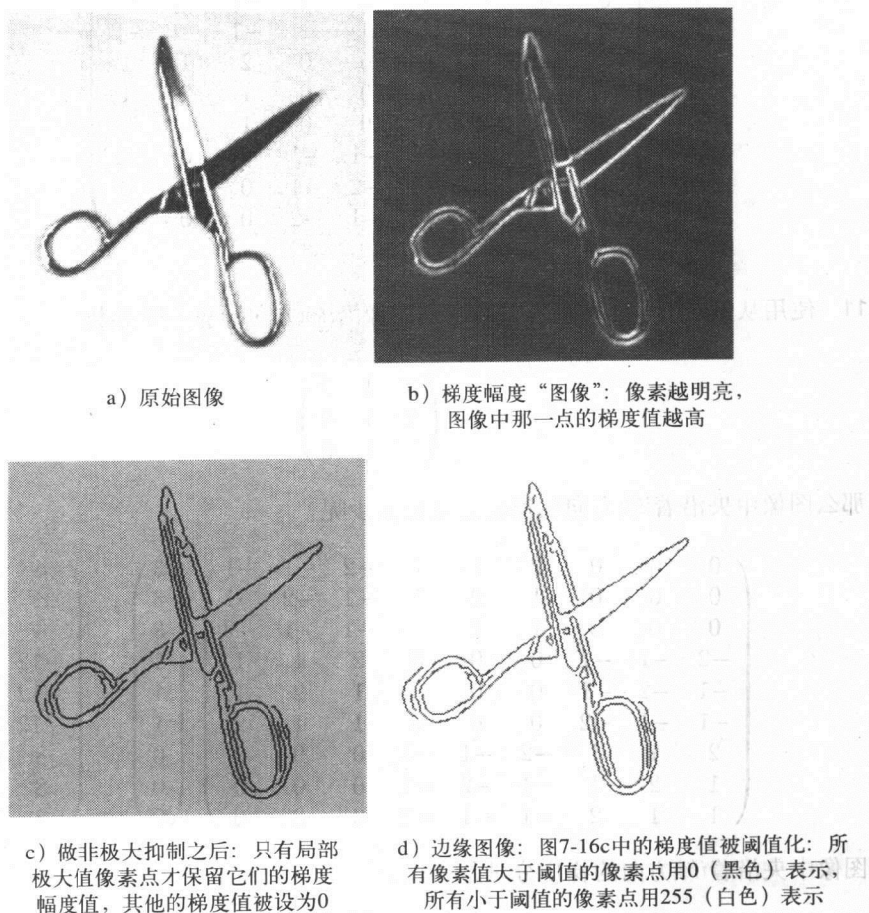


图 7-16

例7.12 给定下列图像

0	1	2	0	4	5
0	0	1	1	4	5
0	2	0	4	5	4
0	0	5	4	6	6
0	0	6	6	5	6
5	4	6	5	4	5

试用下列模板对图像除边界处之外的所有像素点位置上的局部梯度的大小和方向进行估算:

302
303

-1	0	1
-3	0	3
-1	0	1

-1	-3	-1
0	0	0
1	3	1

然后指出哪些代表水平边缘的像素点, 哪些代表垂直边缘的像素点。

当用第一个模板与图像做卷积运算的时候, 便会得到沿着 x (水平) 轴方向的梯度分量的估算:

	5	4	16	17	
	6	11	19	6	
	21	20	7	6	
	24	23	-4	2	

 Δ_x

当用第二个模板与图像做卷积运算的时候, 便会得到沿着 y (垂直) 轴方向的梯度分量的估算:

	1	-1	11	6	
	4	15	15	10	
	0	18	12	4	
	18	8	2	-6	

 Δ_y

梯度的大小为: $|G| = \sqrt{(\Delta_x)^2 + (\Delta_y)^2}$

	$\sqrt{26}$	$\sqrt{17}$	$\sqrt{377}$	$\sqrt{325}$	
	$\sqrt{52}$	$\sqrt{346}$	$\sqrt{586}$	$\sqrt{136}$	
	$\sqrt{441}$	$\sqrt{724}$	$\sqrt{193}$	$\sqrt{52}$	
	$\sqrt{900}$	$\sqrt{593}$	$\sqrt{20}$	$\sqrt{40}$	

(7-25)

304 梯度的方向为:

$$\theta = \tan^{-1} \frac{\Delta I_y}{\Delta I_x}$$

	$\tan^{-1} \frac{1}{5}$	$-\tan^{-1} \frac{1}{4}$	$\tan^{-1} \frac{11}{16}$	$\tan^{-1} \frac{6}{17}$	
	$\tan^{-1} \frac{2}{3}$	$\tan^{-1} \frac{15}{11}$	$\tan^{-1} \frac{15}{19}$	$\tan^{-1} \frac{5}{3}$	
	$\tan^{-1} 0$	$\tan^{-1} \frac{9}{10}$	$\tan^{-1} \frac{12}{7}$	$\tan^{-1} \frac{2}{3}$	
	$\tan^{-1} \frac{3}{4}$	$\tan^{-1} \frac{8}{23}$	$-\tan^{-1} \frac{1}{2}$	$-\tan^{-1} 3$	

我们知道, 对于一个大小在 0° 和 45° 之间($0^\circ \leq |\theta| \leq 45^\circ$)的角度 θ , 它的正切满足 $0^\circ \leq |\tan \theta| \leq 1$ 。如果 $1 < |\tan \theta| \leq \infty$, 那么角度 θ 必定落在范围 45° 到 90° 之间($45^\circ < \theta \leq 90^\circ$)。

由于我们想将梯度方向量化成只有垂直和水平两个方向, 所以可以把所有满足 $0^\circ \leq \theta < 45^\circ$ 的方向看成是水平的, 即设它们的值为0; 而把满足 $45^\circ < \theta \leq 90^\circ$ 的方向看成垂直的, 即设它们的值为 90° 。

通过查找(7-25)所示的这组方向, 可以推导出下面这个梯度方向图:

	0	0	0	0	
	0	90°	0	90°	
	0	0	90°	0	
	0	0	0	90°	

一个像素点是一个水平边缘像素点, 如果它与其垂直邻域的像素做比较时, 它的梯度大小是局部极大值, 同时它的方向是0; 而一个像素点是垂直边缘像素点, 如果它的方向是 90° , 而它的大小在水平方向是局部最大的。

通过查找(7-25)中梯度大小以及我们推导出的量化的方向, 可以确定下列水平(-)边缘:

			-		
		-			

305

例7.13 例7.12所使用的模板是可分离的吗? 如何利用二维模板的可分离性降低卷积的计算量?

例7.12所使用的模板是可分离的, 因为它们可以用两个一维滤波器一个接一个层叠处理

得到:

$$\begin{bmatrix} -1 & 0 & 1 \\ -3 & 0 & 3 \\ -1 & 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix} \quad \text{然后} \quad \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -3 & -1 \\ 0 & 0 & 0 \\ 1 & 3 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 3 & 1 \end{bmatrix} \quad \text{然后} \quad \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

任何二维 $N \times N$ 可分离模板可以用两个大小为 N 的一维模板层叠处理产生。这种做法使得原来每个像素做 N^2 次乘法和加法减少为 $2N$ 次。

7.28 Sobel模板是否对所有图像都适用

Sobel模板对所有带有少量噪声的图像都适用，而对存在大量噪声的图像是不适用的。参见图7-17，图中显示了用Sobel模板对一幅存在大量噪声的图像进行边缘检测的结果。

7.29 如果因为图像中存在着明显的噪声而需要选择一个更大的模板，我们要如何选择权重

下面将考虑如何检测类似图7-18所示的一维信号的突变位置。

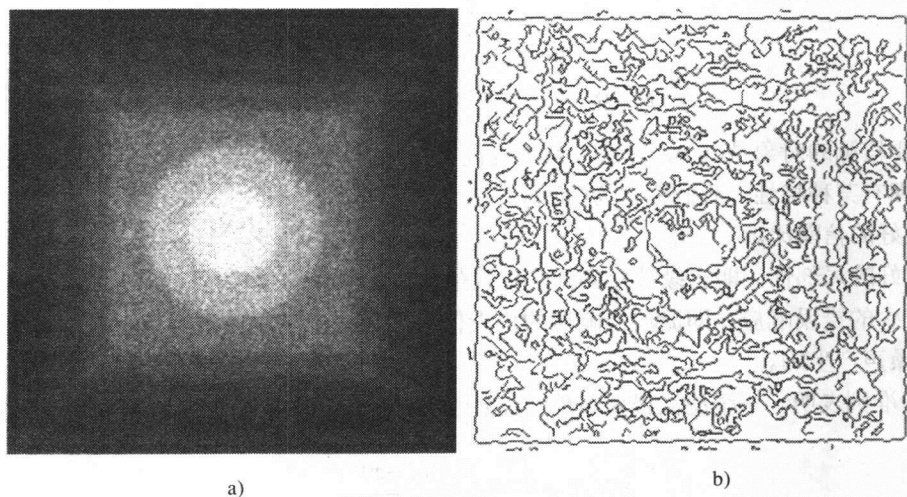


图7-17 使用Sobel边缘检测模板检测一幅模糊的，有严重噪声的图像是非常困难的。

使用不同的阈值，结果都不能让人满意，三个检测结果是在同一个边缘检测框架

下面使用不同的阈值得到的，它们是用不同阈值得到的结果中最好的

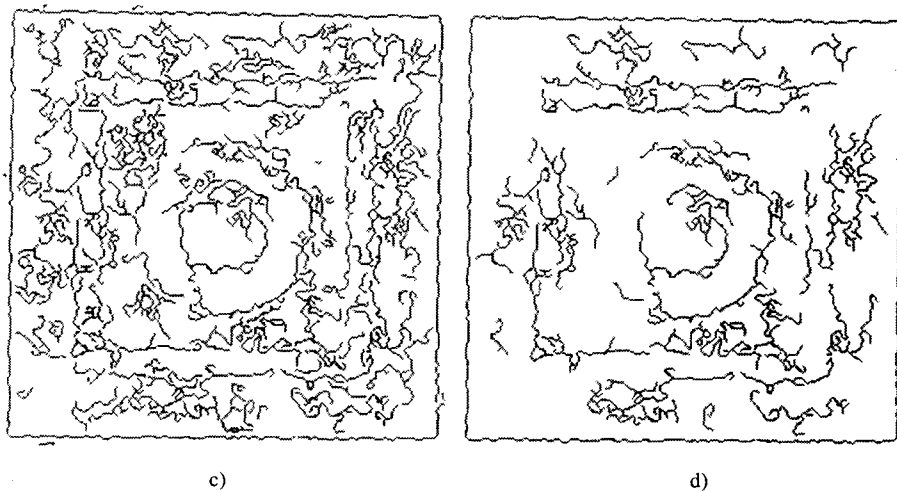


图 7-17 (续)

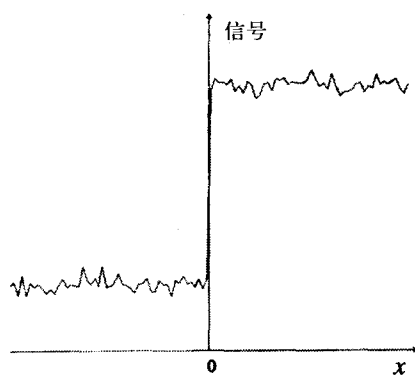


图7-18 一个被噪声污染的一维边缘

假设要检测的对象是 $u(x)$ ，它被大量加性高斯白噪声 $n(x)$ 所淹没，那么用来做边缘检测的模板必须拥有某种特性，我们称之为Canny准则：

- 1) 良好的信噪比。
- 2) 良好的局部性，即边缘被检测到的位置是它实际所在的位置。
- 3) 较少的错误响应，即滤波器响应的极大值大多数情况下应该是主要由真实边缘产生的而不是由噪声产生的。

Canny准则表明，一个滤波器函数 $f(x)$ 有最大的信噪比，如果它使下式的值最大：

$$SNR = \frac{\int_{-\infty}^{\infty} f(x)u(-x)dx}{n_0 \sqrt{\int_{-\infty}^{\infty} f^2(x)dx}} \quad (7-26)$$

其中， n_0 是高斯加性噪声的标准差。由于对一幅特定图像来说，它是一个常量，因此在求解 $f(x)$ 时把它省略以减少计算量。于是，仅需要使下式的值最大：

$$S \equiv \frac{\int_{-\infty}^{\infty} f(x)u(-x)dx}{\sqrt{\int_{-\infty}^{\infty} f^2(x)dx}}$$

Canny准则也表明, 滤波器函数 $f(x)$ 检测的边缘跟实际位置具有最小的偏差, 如果它的一阶导数和二阶导数使得下式的值最大:

$$L \equiv \frac{\int_{-\infty}^{\infty} f''(x)u(-x)dx}{\sqrt{\int_{-\infty}^{\infty} [f'(x)]^2 dx}} \quad (7-27)$$

最后, Canny准则表明, 信号同滤波器 $f(x)$ 做卷积得到的结果包含最少的假警报, 如果函数 $f(x)$ 的一阶导数和二阶导数使得下式的值最大:

306
308

$$C \equiv \sqrt{\frac{\int_{-\infty}^{\infty} (f'(x))^2 dx}{\int_{-\infty}^{\infty} (f''(x))^2 dx}} \quad (7-28)$$

综上所述, 可以设计最优边缘增强滤波器使得上述三个指标最大:

$$S, L, C$$

Canny把前面两个指标合成为一个指标 P , 在约束“ $C = \text{常量}$ ”的条件下最大化该指标, 用这种方法推导出一个卷积滤波器, 但是他自己从来没用过, 因为他注意到可以从高斯函数的导数逼近它, 于是, 他便使用高斯函数的导数作为较好的边缘增强滤波器。

当然, 或者也可以合并三个量 S 、 L 和 C 成为一个指标 P :

$$P \equiv (S \times L \times C)^2 = \frac{\left[\int_{-\infty}^{\infty} f(x)u(-x)dx \right]^2 \left[\int_{-\infty}^{\infty} f''(x)u(-x)dx \right]^2}{\int_{-\infty}^{\infty} f^2(x)dx \int_{-\infty}^{\infty} (f''(x))^2 dx} \quad (7-29)$$

设法寻找 $f(x)$ 使 P 最大, 而出现在 $f(x)$ 的函数表达式中的自由参数可以利用 $f(x)$ 的边界条件计算确定, 再代入 P 表达式中, 选择合适的数值使得 P 值最大。

需要强调的是, 用上述方法定义的滤波器适用于反对称的图像特征 (antisymmetric feature) 的检测, 比如信号中的噪声是加性噪声、白噪声和高斯噪声情况下的阶跃边缘(step edge)或者斜边缘(ramp edge)。

7.30 能否用边缘检测的最优滤波器以最优的方式检测图像中的线

不能。边缘检测滤波器对线的响应是很差的, 因为最优边缘检测滤波器理论是在假设图像中只有单独一个阶跃边缘而推导出来的。可以从式(7-26)使用的积分的范围看出这一点。例如, 积分范围是从 $-\infty$ 到 $+\infty$, 假设整个无限长的信号区间上只有一个阶跃边缘。如果有线存在的话, 它的外形会如图7-19所示。

该图看起来像两个相反方向的阶跃边缘靠在一起, 滤波器对这两个阶跃边缘的响应交错在一起, 因此效果并不令人满意: 两个阶跃的地方可能都检测得到, 也可能检测不到。假如都被检测到了, 这两个边缘也会很容易与真实位置错位, 并且两个边缘的相对位置偏移。

309 除了上述的差异,阶跃边缘和线还有一个更基本的差别。

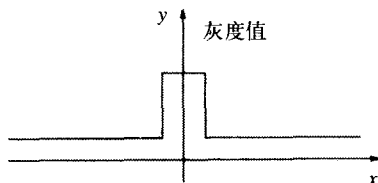
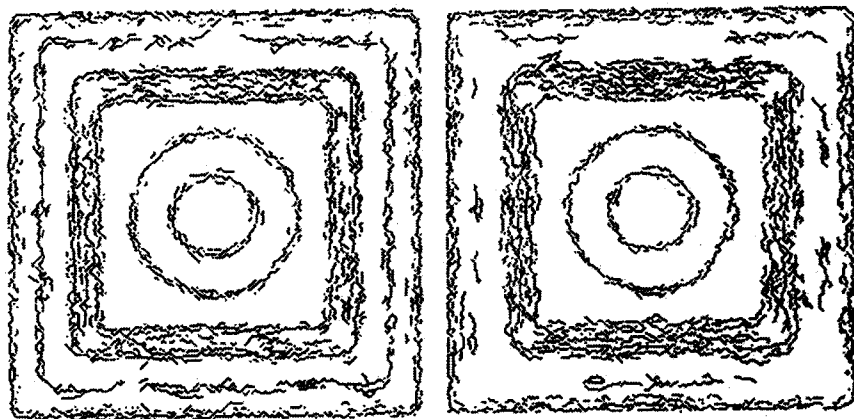


图7-19 线的轮廓

7.31 阶跃边缘和线的基本差别是什么

阶跃边缘是尺度不变的: 无论是拉伸还是收缩, 阶跃边缘看起来都是一样的; 而线有固定长度的尺度: 即它的宽度。一旦要检测的特征有了自己的“尺寸”, 滤波器的尺寸就变得重要了。如果想推导出适合检测斜边缘而不是阶跃边缘的滤波器, 同样的问题必须考虑: 斜坡上升(或者下降)时跨越的长度就是该特征的特性, 在设计最优滤波器时不能够忽略。Petrou和Kittler使用式(7-29)中的准则, 经过适当的修改, 推导出一个适合各种斜率的斜边缘检测最优滤波器。图7-20显示的是试图用错误的滤波器检测图7-17a中的图像, 图像的边缘被模糊了, 实际上, 该边缘有像斜坡一样的外形, 而使用的滤波器是阶跃边缘检测最优滤波器。图7-21显示的是在同一幅图像上使用斜边缘检测最优滤波器, 及不同的滤波器尺寸对结果的影响。图7-22显示的是对一幅相对“清晰”且“已去模糊”的图像做边缘检测的结果, 在这种情况下, 不必太在意使用什么样的滤波器。

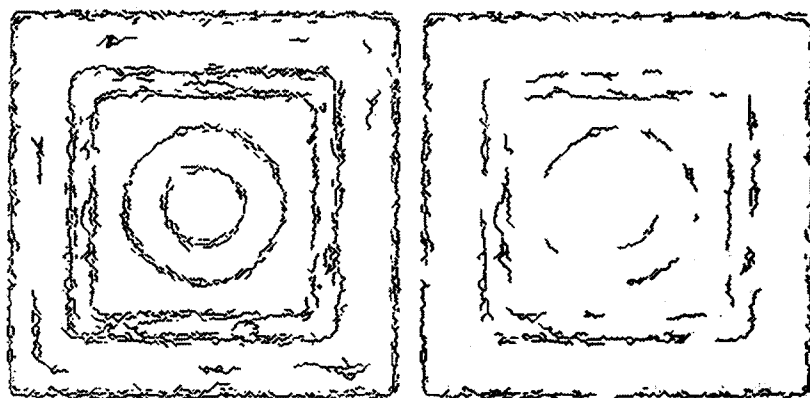
Petrou针对检测线的情况也修改了Canny准则, 并且得到了线检测的最优滤波器, 它是依赖于线的宽度和锐化的程度的。



a) 假定有一个阶跃边缘并用尺寸为
17 × 17的滤波器进行滤波

b) 与a) 相同的参数, 只是滤波器
尺寸扩大为21 × 21, 一部分细节
丢失, 效果并没有改善

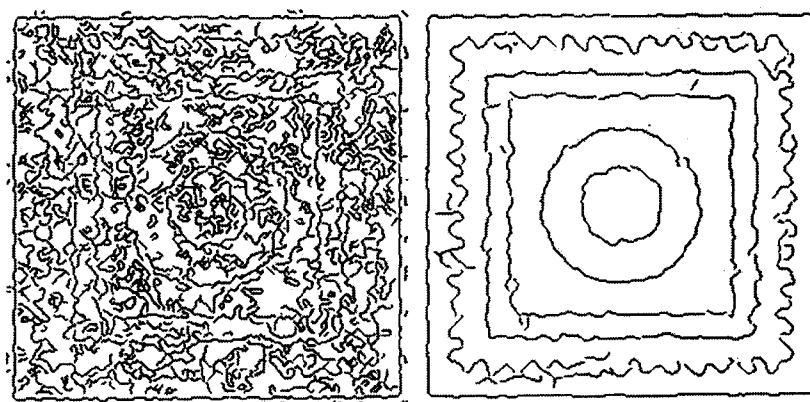
图7-20 如果边缘被模糊了且类似斜线形状, 则使用大尺寸的最优滤波器处理高层次的噪声, 同时采用阈值化的方法似乎不起作用。这个“最优”滤波器对于阶跃边缘是最优的



c) 假定有一个阶跃边缘, 用尺寸为 17×17 的滤波器进行滤波, 并且阈值增大

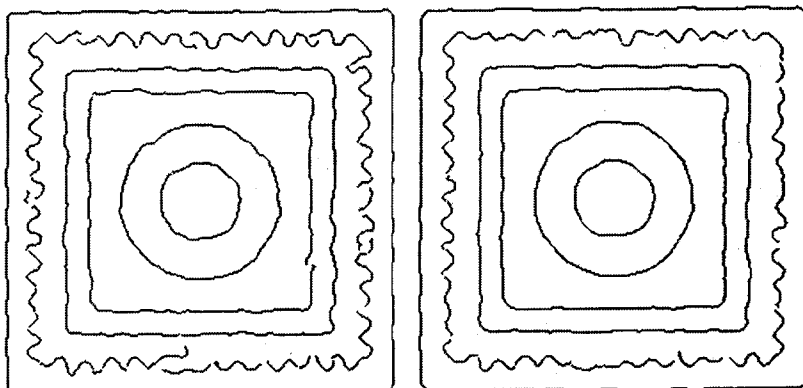
d) 与 c) 相同, 但使用更大的阈值

图 7-20 (续)



a) 假定是一个窄的斜边缘, 并用尺寸为 7×7 的滤波器进行滤波

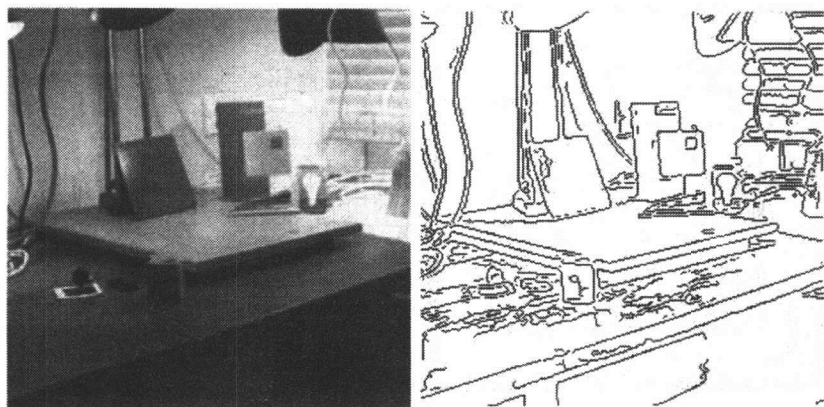
b) 用与 a) 同样的方法, 只是设置滤波器尺寸为 13×13



c) 假定是一个斜边缘, 并用尺寸为 17×17 的滤波器进行滤波

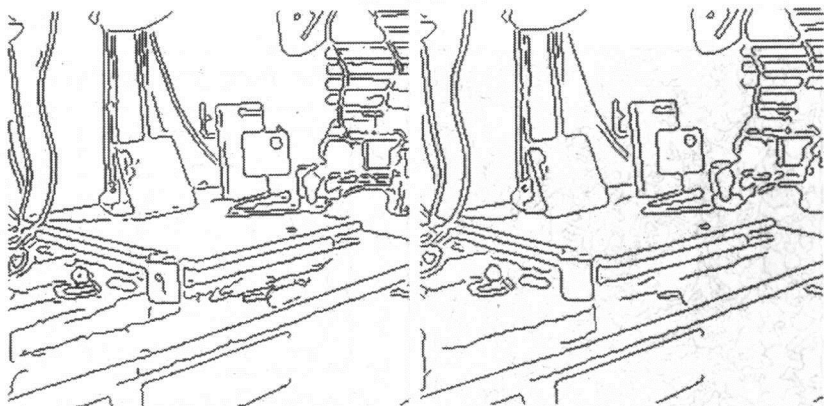
d) 假定是一个斜边缘, 并用尺寸为 21×21 的滤波器进行滤波

图7-21 对要检测的特征选择了正确的模型, 针对该层次的噪声使用合适的滤波器尺寸被证明是至关重要的



a) 原始图像

b) 使用Sobel滤波器的结果



c) 使用尺寸为5×5的最优滤波器的结果

d) 使用尺寸为7×7的最优滤波器的结果

图7-22 日常生活中的图像相对来说清晰，所以使用尺寸相对小的滤波器就可以得到很好的结果，注意，滤波器尺寸越小，能保留越多的细节

- 例7.14(B) 假设定义 $u(x)$ 中的 x 可以取正值和负值，我们要增强 $u(x)$ 在 $x=0$ 位置的特征。由式(7-26)可以知道，当要增强的特征使得 $u(x)$ 成为一个偶函数，我们必须选择偶函数滤波器；当要增强的特征使得 $u(x)$ 成为一个奇函数，我们必须选择奇函数滤波器。

任何函数 $f(x)$ 都可以写成一个偶函数和一个奇函数的和：

$$f(x) \equiv \underbrace{\frac{1}{2}[f(x) - f(-x)]}_{f_o(x) \text{ (奇)}} + \underbrace{\frac{1}{2}[f(x) + f(-x)]}_{f_e(x) \text{ (偶)}}$$

于是， $f(x)$ 写成：

$$f(x) = f_e(x) + f_o(x)$$

假设 $u(x)$ 是偶函数，那么 S 的分子中的积分就是：

$$\begin{aligned}\int_{-\infty}^{\infty} f(x)u(-x)dx &= \int_{-\infty}^{\infty} f_e(x)u(-x)dx + \underbrace{\int_{-\infty}^{\infty} f_o(x)u(-x)dx}_{\substack{\text{奇函数在一个对称区间} \\ \text{内积分，结果为0}}} \\ &= \int_{-\infty}^{\infty} f_e(x)u(-x)dx\end{aligned}$$

表达式 S 中的分母的积分是：

$$\begin{aligned}\int_{-\infty}^{\infty} f^2(x)dx &= \int_{-\infty}^{\infty} f_e^2(x)dx + \int_{-\infty}^{\infty} f_o^2(x)dx + \underbrace{2 \int_{-\infty}^{\infty} f_e(x)f_o(x)dx}_{\substack{\text{奇函数在一个对称区间} \\ \text{内积分，结果为0}}} \\ &= \int_{-\infty}^{\infty} f_e^2(x)dx + \int_{-\infty}^{\infty} f_o^2(x)dx\end{aligned}$$

因此，可以看出在整个信号的响应中，滤波器的奇函数部分根本没起什么作用，相反，它产生了噪声响应：也就是降低了信号的信噪比。因此，要想增强偶特征，必须使用偶函数滤波器；同样，要想增强奇特征，必须使用奇函数滤波器。

例7.15 下列数字滤波模板适合于增强哪种类型的特征？

-1	2	-1
-1	2	-1
-1	2	-1

-1	-1	-1
2	2	2
-1	-1	-1

第一个滤波器适合增强垂直线而第二个适合增强水平线，两种情况下都是线的亮度大于背景。

例7.16 使用例7.16中的模板处理下列图像（不处理边界像素）。

1	1	5	3	0
0	1	4	1	0
1	1	3	2	1
0	2	5	3	0
1	0	4	2	0

然后，对计算得到的输出图像选择合适的阈值，构造出输入图像的特征图。

图像与第一个模板卷积的结果是：

	-8	15	-1	
	-5	14	-1	
	-8	14	1	

图像与第二个模板卷积的结果是:

	-2	-3	-4	
	-2	-4	-1	
	4	8	4	

(对边界像素没有输出。)

注意到输出结果包含的值很容易地分为两类: 正的和负的。于是选择值 $t = 0$ 作为阈值, 该阈值似乎处于两类输出值总体最大间距处 (即, 一类总体大致表示背景, 另一类总体表示要检测的特征)。确定阈值后, 给一类标上一个标记 (也就是将所有大于阈值的值标上1), 给另外一类标上另一个标记 (也就是将所有小于阈值的值标上0)。于是, 两类输出可以表示为:

0	1	0
0	1	0
0	1	0

0	0	0
0	0	0
1	1	1

这就是给定图像的特征图。

B7.3 一个随机噪声信号与一个滤波器进行卷积运算。

假设噪声信号是 $n(x)$, 让它与滤波器 $f(x)$ 进行卷积运算卷积, 卷积的结果是:

$$g(x_0) = \int_{-\infty}^{\infty} f(x)n(x_0 - x)dx \quad (7-30)$$

既然输入的噪声是一个随机过程, 上式的结果也应该是个随机过程。如果我们取它的期望值, 它的值应该和输入的噪声的期望值一样都为0。为了刻画该噪声, 我们计算它的方差, 也就是均方值, 由下式得到:

$$E\{[g(x_0)]^2\} = E\{g(x_0)g(x_0)\} \quad (7-31)$$

这其实是输出的自相关函数当参数为0时的情况, 因此, 先得计算 g 的自相关 $R_{gg}(0)$,

在式(7-30)两边都乘上 $g(x_0 + \tau)$ 并取均值:

$$\begin{aligned}
 E\{g(x_0)g(x_0 + \tau)\} &= E\left\{\int_{-\infty}^{\infty} f(x)g(x_0 + \tau)n(x_0 - x)dx\right\} \\
 \Rightarrow R_{gg}(\tau) &= \int_{-\infty}^{\infty} f(x)E\{g(x_0 + \tau)n(x_0 - x)\}dx \\
 \Rightarrow R_{gg}(\tau) &= \int_{-\infty}^{\infty} f(x)R_{gn}(x_0 + \tau - x_0 + x)dx \\
 \Rightarrow R_{gg}(\tau) &= \int_{-\infty}^{\infty} f(x)R_{gn}(\tau + x)dx
 \end{aligned} \tag{7-32}$$

其中, $R_{gn}(a)$ 是输入噪声信号和输出噪声信号在相对平移量为 a 时的互相关函数。因此, 必须先计算 R_{gn} , 在式(7-30)两边都乘上 $n(x_0 - \tau)$ 再取均值。(这里乘上 $n(x_0 - \tau)$ 而不是 $n(x_0 + \tau)$ 是因为在式(7-32)中我们定义 R_{gn} 的参数为 g 的参数减去 n 的参数。

$$\begin{aligned}
 E\{g(x_0)n(x_0 - \tau)\} &= \int_{-\infty}^{\infty} f(x)E\{n(x_0 - x)n(x_0 - \tau)\}dx \\
 \Rightarrow R_{gn}(\tau) &= \int_{-\infty}^{\infty} f(x)R_{nn}(x_0 - x - x_0 + \tau)dx \\
 \Rightarrow R_{gn}(\tau) &= \int_{-\infty}^{\infty} f(x)R_{nn}(\tau - x)dx
 \end{aligned} \tag{7-33}$$

其中, $R_{nn}(a)$ 是输入噪声信号在相对平移量为 a 时的自相关函数。但是, $n(x)$ 是高斯白噪声, 它的自相关函数是一个由 $R_{nn}(\tau) = n_0^2 \delta(\tau)$ 表示的delt函数, 其中 n_0^2 是噪声的方差。于是得到:

$$R_{gn}(\tau) = \int_{-\infty}^{\infty} f(x)n_0^2 \delta(\tau - x)dx = n_0^2 f(\tau) \tag{7-34}$$

故, 如式(7-32)所示, R_{gn} 在参数为 $\tau + x$ 时有:

$$R_{gn}(\tau + x) = n_0^2 f(\tau + x) \tag{7-35}$$

代入式(7-32)中有:

$$R_{gg}(\tau) = n_0^2 \int_{-\infty}^{\infty} f(x)f(\tau + x)dx \Rightarrow R_{gg}(0) = n_0^2 \int_{-\infty}^{\infty} f^2(x)dx \tag{7-36}$$

再代入式(7-31)中, 得到:

$$E\{[g(x_0)]^2\} = n_0^2 \int_{-\infty}^{\infty} f^2(x)dx \tag{7-37}$$

上式的平方根便是滤波器对噪声的平均响应。

B7.4 计算带噪声的边缘信号与滤波器卷积后的信噪比

假如 $f(x)$ 是要用于推导的滤波器，它能增强带噪声信号中的边缘。该信号包含两个成分：我们感兴趣的确定信号 $u(x)$ 和随机噪声成分 $n(x)$ 为：

317

$$I(x) = u(x) + n(x)$$

当有噪信号与滤波器进行卷积的时候，来源于有噪信号确定部分的响应为：

$$s(x_0) = \int_{-\infty}^{\infty} f(x)u(x_0 - x)dx \quad (7-38)$$

假设要检测边缘的实际位置为 $x_0 = 0$ ，于是：

$$s_0 = \int_{-\infty}^{\infty} f(x)u(-x)dx \quad (7-39)$$

来源于噪声部分（见方框B7.3）的平均滤波响应是 $n_0 \sqrt{\int_{-\infty}^{\infty} f^2(x)dx}$ ，故信噪比为：

$$SNR = \frac{\int_{-\infty}^{\infty} f(x)u(-x)dx}{n_0 \sqrt{\int_{-\infty}^{\infty} f^2(x)dx}} \quad (7-40)$$

B7.5 好的局部性指标的推导

边缘的位置是由输出极大的那些点所标识出来的，卷积的全部输出是由确定信号 $u(x)$ 的输出和噪声成分 $n(x)$ 的输出组成的。假如有噪信号为：

$$I(x) = u(x) + n(x)$$

它与滤波器 $f(x)$ 卷积的结果为：

$$\begin{aligned} O(x_0) &\equiv \int_{-\infty}^{\infty} f(x)I(x_0 - x)dx = \int_{-\infty}^{\infty} f(x_0 - x)I(x)dx \\ &= \int_{-\infty}^{\infty} f(x_0 - x)u(x)dx + \int_{-\infty}^{\infty} f(x_0 - x)n(x)dx \equiv s(x_0) + g(x_0) \end{aligned}$$

318

其中， $s(x_0)$ 是我们感兴趣的确定信号的输出，而 $g(x_0)$ 是噪声的输出。

边缘是在该输出的局部极大值的地方被检测到的，即，在那些对 x_0 的一阶导数为0的地方：

$$\frac{dO(x_0)}{dx_0} = \frac{ds(x_0)}{dx_0} + \frac{dg(x_0)}{dx_0} = \int_{-\infty}^{\infty} f'(x_0 - x)u(x)dx + \int_{-\infty}^{\infty} f'(x_0 - x)n(x)dx \quad (7-41)$$

在点 $x_0 = 0$ ，也就是边缘的实际位置，展开滤波器的一阶导数，并保留展开式的前两项：

$$f'(x_0 - x) \simeq f'(-x) + x_0 f''(-x) + \dots$$

代入 $\frac{ds(x_0)}{dx_0}$ 得到：

$$\frac{ds(x_0)}{dx_0} \simeq \int_{-\infty}^{\infty} f'(-x)u(x)dx + x_0 \int_{-\infty}^{\infty} f''(-x)u(x)dx \quad (7-42)$$

我们所要检测的边缘具有反对称的特征，即其边缘的形状类似 $\text{—}\square\text{—}$ 。

根据例7.14证明过的结论， $f(x)$ 也必须是个反对称函数。这意味着它的一阶导数是对称的，且当它与一个反对称函数 $u(x)$ 相乘时，会得到一个反对称的被积函数。由于积分区间是对称的，因此式(7-42)的第一项为0。也就是：

$$\frac{ds(x_0)}{dx_0} \simeq \underbrace{x_0 \int_{-\infty}^{\infty} f''(-x)u(x)dx}_{\text{记 } \tilde{x} = -x} = x_0 \int_{-\infty}^{\infty} f''(\tilde{x})u(-\tilde{x})d\tilde{x} \quad (7-43)$$

而滤波器对噪声响应的微分可以用一种更简便的方法写出来：

$$\frac{dg(x_0)}{dx_0} = \int_{-\infty}^{\infty} f'(x_0 - x)n(x)dx = \int_{-\infty}^{\infty} f'(x)n(x_0 - x)dx \quad (7-44)$$

边缘的位置由下式 x 的值给出，这个值使得式(7-43)和式(7-44)右边的和为0：

$$\begin{aligned} \frac{ds(x_0)}{dx_0} + \frac{dg(x_0)}{dx_0} &= 0 \Rightarrow \\ x_0 \int_{-\infty}^{\infty} f''(x)u(-x)dx + \int_{-\infty}^{\infty} f'(x)n(x_0 - x)dx &= 0 \Rightarrow \\ x_0 \int_{-\infty}^{\infty} f''(x)u(-x)dx &= - \int_{-\infty}^{\infty} f'(x)n(x_0 - x)dx \end{aligned} \quad (7-45)$$

该式的右边是一个随机变量，表明边缘的位置由它的真正位置（也就是在 $x_0 = 0$ 处）的随机分布位置确定。我们可以通过计算 x_0 的均值表达相对于真实位置的平均偏移量，但是这个值应该是0。因此我们转而计算 x_0 的方差，对式(7-45)两边平方再求均值得到：

$$E\{x_0^2\} \left[\int_{-\infty}^{\infty} f''(x)u(-x)dx \right]^2 = E \left\{ \left[\int_{-\infty}^{\infty} f'(x)n(x_0 - x)dx \right]^2 \right\}$$

注意，求均值算子只对式子中的随机成分起作用。

在方框B7.3中, 我们知道如果一个方差为 n_0^2 的噪声信号与滤波器 $f(x)$ 做卷积, 输出信号的均方值是:

$$n_0^2 \int_{-\infty}^{\infty} f^2(x) dx$$

(参见式(7-37))。式(7-45)的右边表示噪声成分与滤波器 $f'(x)$ 的卷积, 因而它的均方值是 $n_0^2 \int_{-\infty}^{\infty} [f'(x)]^2 dx$, 于是有:

$$E\{x_0^2\} = \frac{n_0^2 \int_{-\infty}^{\infty} [f'(x)]^2 dx}{\left[\int_{-\infty}^{\infty} f''(x) u(-x) dx \right]^2} \quad (7-46)$$

该期望值越小, 边缘的局部化就越好。于是可以将好的局部性度量定义为上式平方根的倒数; 同时, 还可以再忽略因子 n_0 , 因为它是噪声的标准差, 在图像处理过程中, 我们对该量不做控制。因此, 一个滤波器对好的局部性是最优的, 如果它能使下面的量最大:

$$L \equiv \frac{\int_{-\infty}^{\infty} f''(x) u(-x) dx}{\sqrt{\int_{-\infty}^{\infty} [f'(x)]^2 dx}} \quad (7-47)$$

例7.17(B) 说明一个信号 $u(x)$ 和一个滤波器 $f(x)$ 做卷积运算的输出的微分可以通过信号与滤波器的导数做卷积运算得到。

卷积的输出是:

$$s(x_0) = \int_{-\infty}^{\infty} f(x) u(x_0 - x) dx$$

或者

$$s(x_0) = \int_{-\infty}^{\infty} f(x_0 - x) u(x) dx \quad (7-48)$$

应用Leibnitz准则, 对积分中的参数求微分 (方框B7.1), 从式(7-48)可以得到:

$$\frac{ds(x_0)}{dx_0} = \int_{-\infty}^{\infty} f'(x_0 - x) u(x) dx$$

做变量替换得到:

$$\frac{ds(x_0)}{dx_0} = \int_{-\infty}^{\infty} f'(x)u(x_0 - x)dx$$

B7.6 最大错误量的推导。

Rice曾经证明，一个函数 h 和高斯噪声做卷积运算后的零交叉的平均密度是：

$$x_{av} = \pi \left[\frac{-R_{hh}(0)}{R''_{hh}(0)} \right]^{\frac{1}{2}} \quad (7-49)$$

其中， $R_{hh}(\tau)$ 是函数 $h(x)$ 在空域的自相关函数，即：

$$R_{hh}(\tau) = \int_{-\infty}^{\infty} h(x)h(x+\tau)dx \quad (7-50)$$

于是有：

$$R_{hh}(0) = \int_{-\infty}^{\infty} (h(x))^2 dx$$

使用Leibnitz准则（见方框B7.1），在式(7-50)中对 τ 求微分得到：

$$R'_{hh}(\tau) = \int_{-\infty}^{\infty} h(x)h'(x+\tau)dx$$

定义一个新的积分变量 $\tilde{x} = x + \tau \Rightarrow x = \tilde{x} - \tau$ 同时 $d\tilde{x} = dx$ ，可得：

$$R'_{hh}(\tau) = \int_{-\infty}^{\infty} h(\tilde{x} - \tau)h'(\tilde{x})d\tilde{x} \quad (7-51)$$

再做一次微分（式(7-51)）：

$$R''_{hh}(\tau) = - \int_{-\infty}^{\infty} h'(\tilde{x} - \tau)h'(\tilde{x})d\tilde{x} \Rightarrow R''_{hh}(0) = - \int_{-\infty}^{\infty} (h'(x))^2 dx$$

故一个噪声信号用函数 $h(x)$ 滤波，输出信号的零交叉的平均间隔是：

$$x_{av} = \pi \sqrt{\frac{\int_{-\infty}^{\infty} (h(x))^2 dx}{\int_{-\infty}^{\infty} (h'(x))^2 dx}} \quad (7-52)$$

从方框B7.5分析可以看出，在我们这种情况下，最大错误量可从等式 $\int_{-\infty}^{\infty} f'(x)n(x_0 - x)dx = 0$

得来, 该式是没有任何信号($u(x) = 0$)情况下的错误响应, 它相当于当噪声信号与函数 $f'(x)$ 做卷积时输出信号中最大错误量为0。因此, 如果要降低错误的局部极大值数, 对滤波器函数 $f'(x)$ 来说, 必须尽可能增加零交叉点之间的平均距离, 因而我们用下式作为假警报数量的好的度量指标:

$$C \equiv \sqrt{\frac{\int_{-\infty}^{\infty} (f'(x))^2 dx}{\int_{-\infty}^{\infty} (f''(x))^2 dx}}$$

7.32 本章要点

本章解决了一幅图像中的信息内容的降维问题, 使得计算机视觉系统可以更简单地进行处理。为了实现这个目的, 本章讨论了两种基本方法: 区域分割和边缘检测。区域分割是尽可能将图像中具有空间相同特征的像素块从空间上分开, 这些像素块可能代表着图像描述的物体的表面或者表面的一部分; 边缘检测的目的是尽可能检测到上述一致区域的边界。最普通的边缘检测方法是基于图像一阶导数的估算, 对于具有少量噪声的图像, Sobel模板可以用来增强边缘; 对于有严重噪声的图像, 我们应该使用Canny滤波器。Canny滤波器可以用一

个高斯函数的导数来逼近, 即它们的形式是 $xe^{-\frac{x^2}{2\sigma^2}}$ 。根据Canny准则, 为了使得滤波器最优, 参数 σ 在表达式中必须有一个确定的值, 但是, 人们往往把 σ 当作一个自由参数并改变它的值进行试验。在离散化滤波器和做高斯函数截断的时候必须多加注意, 千万不要构造两端带突起的滤波器。无论是使用Sobel模板还是高斯函数的导数, 结果都是增强图像中的边缘, 输出结果还要使用非极大抑制进行处理 (即在输出结果中确认局部极大值), 然后再确定阈值 (保留有意义的局部极大值)。

边缘检测器包含上述三个步骤, 最后得到的是边缘的片段, 通常还要有进一步的处理, 那就是把片段连接起来, 构造出封闭的轮廓来标识连续一致的区域; 有时人们也可能跳过这一步直接进行区域分割, 如果需要, 再提取区域的边界。在确定像素点所属区域时, 如果同时考虑属性的相似性以及空间的近邻关系, 那么基于区域的方法会更加有效。

参考文献

本书的素材大部分来源于经典教科书。例6.10所述的方法取自文献3。第7章主要是基于研究论文的内容写的。在下面的文献中，文献13、15和23与阈值方法有关；文献5、6、13、18-22和28与线性边缘检测有关；论文17与非线性边缘检测有关，它用到了论文10介绍的方法；文献21是一篇关于将Canny相关方法应用于边缘检测的综述。

- 1 M Abramowitz and I A Stegun (eds), 1970. *Handbook of Mathematical Functions*, Dover Publications, ISBN 0-486-61272-4.
- 2 K G Beauchamp, 1975. *Walsh Functions and their Applications*, Academic Press, ISBN 0-12-084050-2.
- 3 C R Boukouvalas, 1996. *Colour Shade Grading and its Applications to Visual Inspection*. PhD thesis, University of Surrey, UK.
- 4 R N Bracewell, 1978. *The Fourier Transform and its Applications*, McGraw-Hill, ISBN 0-07-007013-X.
- 5 J Canny, 1983. *Finding Edges and Lines in Images*. MIT AI Lab Technical Report 720.
- 6 J Canny, 1986. "A computational approach to edge detection". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol PAMI-8, pp 679-698.
- 7 R A Gabel and R A Roberts, 1987. *Signals and Linear Systems*, John Wiley & Sons, ISBN 0-471-83821-7.
- 8 R C Gonzalez and R E Woods, 1992. *Digital Image Processing*, Addison-Wesley, ISBN 0-201-50803-6.
- 9 I S Gradshteyn and I M Ryzhik, 1980. *Table of Integrals, Series and Products*, Academic Press, ISBN 0-12-294760-6.
- 10 J Graham and C J Taylor, 1988. "Boundary cue operators for model-based image processing". *Proceedings of the Fourth Alvey Vision Conference, AVC88, University of Manchester, 31 August-2 September 1988*, pp 59-64.

- 11 H P Hsu, 1970. *Fourier Analysis*, Simon & Schuster, New York.
- 12 T S Huang (ed), 1979. "Picture processing and digital filtering". In *Topics in Applied Physics*, Vol 6, Springer-Verlag, ISBN 0-387-09339-7.
- 13 J Kittler, 1983. "On the accuracy of the Sobel edge detector". *Image and Vision Computing*, Vol 1, pp 37-42.
- 14 J Kittler and J Illingworth, 1985. "On threshold selection using clustering criteria". *IEEE Transactions on Systems, Man, and Cybernetics*, Vol SMC-15, pp 652-655.
- 15 N Otsu, 1979. "A threshold selection method from gray level histograms". *IEEE Transactions on Systems, Man, and Cybernetics*, Vol SMC-9, pp 62-66.
- 16 A Papoulis, 1965. *Probability, Random Variables and Stochastic Processes*, McGraw-Hill Kogakusha Ltd, Library of Congress Catalog Card number 64-22956.
- 17 I Pitas and A N Venetsanopoulos, 1986. "Non-linear order statistic filters for image filtering and edge detection". *Signal Processing*, Vol 10, pp 395-413.
- 18 M Petrou and J Kittler, 1991. "Optimal edge detectors for ramp edges". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol PAMI-13, pp 483-491.
- 19 M Petrou and A Kolomvas, 1992. "The recursive implementation of the optimal filter for the detection of roof edges and thin lines". In *Signal Processing VI, Theory and Applications*, pp 1489-1492.
- 20 M Petrou, 1993. "Optimal convolution filters and an algorithm for the detection of wide linear features." *IEE Proceedings: Vision, Signal and Image Processing*, Vol 140, pp 331-339.
- 21 M Petrou, 1994. "The differentiating filter approach to edge detection". *Advances in Electronics and Electron Physics*, Vol 88, pp 297-345.
- 22 M Petrou, 1995. "Separable 2D filters for the detection of ramp edges". *IEE Proceedings: Vision, Image and Signal Processing*, Vol 142, pp 228-231.
- 23 M Petrou and A Matrucceli, 1998. "On the stability of thresholding SAR images". *Pattern Recognition*, Vol 31, pp 1791-1796.
- 24 W K Pratt, 1978. *Digital Image Processing*, John Wiley & Sons, ISBN 0-471-01888-0.
- 25 R M Pringle and A A Rayner, 1971. *Generalized Inverse Matrices with Applications to Statistics*, No. 28 of Griffin's Statistical monographs and courses, edited by A Stuart, ISBN 0-85264-181-8.

-
- 26 S O Rice, 1945. "Mathematical analysis of random noise". *Bell Systems Tech. J.*, Vol 24, pp 46-156.
 - 27 A Rosenfeld and A C Kak, 1982. *Digital Picture Processing*, Academic Press, ISBN 0-12-597301-2.
 - 28 L A Spacek, 1986. "Edge detection and motion detection". *Image and Vision Computing*, Vol 4, pp 43-53.
 - 29 P H Winston, 1992. *Artificial Intelligence*, Addison-Wesley, ISBN 0-20-153377-4.
 - 30 T Y Young and K S Fu (eds), 1986. *Handbook of Pattern Recognition and Image Processing*, Academic Press, ISBN 0-12-774560-2.

主题词索引

索引中的页码为英文原书页码, 与书中边栏页码一致

A

additive noise (加性噪声), 144
antisymmetric image features (反对称的图像特征), 309
approximation theory for filter design (滤波器设计的逼近理论), 171 ~ 191
approximation of an image using its K-L transform (图像的K-L变换逼近), 110, 115
approximation of an image by SVD (图像的SVD逼近), 34, 35
autocorrelation function of a random field (随机场的自协方差), 94, 103
autocorrelation matrix of an image (how to compute) (图像的自相关矩阵 (如何计算)), 98 ~ 102, 105 ~ 106
autocovariance of a random field, (随机场的协方差), 94
automatic vision (机器视觉), 18

B

bandpass filter (带通滤波器), 160
bands of an image (图像的频带), 1, 138
barrel distortion (桶形失真), 194
basis images for the Fourier transform (傅里叶变换的基图像), 68, 69
basis images for the Haar transform (哈尔变换的基图像), 52
basis images for the Hadamard transform (哈达玛变换的基图像), 58
basis images from the K-L transform (K-L变换的基图像), 111
basis images for the Walsh transform (沃尔什变换的基图像), 58
bilinear interpolation (双线性插值), 195
bits needed for an image (存储图像所需的比特), 2
block circulant matrix (块循环矩阵), 232, 239 ~ 240
brightness of an image pixel (像素的亮度), 2

C

Canny's criteria for an optimal edge filter (最佳边缘检测滤波器的Canny准则), 306 ~ 309, 318 ~ 322
characteristic equation of a matrix (矩阵的特征方程), 31

Chebyshev norm (Chebyshev 范数), 171, 172
checkerboard effect (棋盘状效果), 2
circulant matrix (循环矩阵), 233
clustering (聚类), 288
complete set of functions (函数的完全集), 45
conditional expectation (条件期望值), 217
constraint matrix inversion (约束矩阵求逆), 251
contrast enhancement of a multispectral image (多光谱图像的对比度增强), 135
contrast manipulation (对比度处理), 132
convolution (卷积), 7, 72 ~ 77, 157
convolution theorem (assumptions of) (卷积定理 (假设)), 75
covariance of two random variables (两个随机变量的协方差), 92
cross correlation of two random fields (两个随机场的互相关性), 95
cross covariance of two random fields (两个随机场的互换方差), 95
cross spectral density of two random fields (两个随机场的互谱密度), 218

D

delta function (delta函数), 8
DFT (see discrete Fourier transform)
direct component of an image (DFT (参看离散傅里叶变换) 图像的直流分量), 82
discrete cosine transform (离散余弦变换), 86
discrete Fourier transform (离散傅里叶变换), 63 ~ 86
discrete Fourier transform (display of) (离散傅里叶变换 (的显示)), 79
discrete Fourier transform and image restoration (离散傅里叶变换和图像复原), 210 ~ 215, 228 ~ 230
discrete Fourier transform of a rotated image (旋转图像的离散傅里叶变换), 79
discrete Fourier transform of a scaled image (伸缩图像的离散傅里叶变换), 83
discrete Fourier transform of a shifted image (位移图像的离散傅里叶变换), 81
distribution function of a random variable (单个随机变量的分布函数), 90
distribution function of many random variables (多个随

机变量的分布函数), 92
dual grid (二重网格), 291

E

edge detection (边缘检测), 265, 289 ~ 322
edge map (边缘图), 303
edge pixels (边缘像素), 291
edgels (边缘像素), 291
eigenimages from the K-L transform (K-L变换的特征图像), 121
eigenimages from SVD (SVD的特征图像), 37
eigenvalues of a circulant matrix (循环矩阵的特征值), 233
eigenvalues of a matrix (矩阵特征值), 30, 140, 234
eigenvectors of a circulant matrix (循环矩阵的特征向量), 233
eigenvectors of a matrix (矩阵的特征向量), 30, 31, 140, 234
ergodic random field (遍历随机场), 96, 97
ergodic random field with respect to the mean (均值遍历随机场), 97
ergodic random field with respect to the autocorrelation function (自相关函数遍历随机场), 97
ergodicity (遍历性) 102, 103, 104, 227
error in approximating an image by its K-L transform (用K-L变换逼近图像的误差), 115
error in approximating an image by its SVD (用SVD逼近图像的误差), 35
even symmetrical cosine transform (偶对称余弦变换), 86
expansion of an image in terms of Fourier basis functions (图像的傅里叶基函数展开), 68
expansion of an image in terms of eigenimages (图像的特征图像展开), 34
expansion of an image in terms of Haar basis functions (图像的哈尔基函数展开), 51 ~ 52, 58 ~ 60
expansion of an image in terms of Walsh/Hadamard matrices (图像的沃尔什/哈达玛矩阵展开), 57 ~ 58, 60 ~ 61
expansion of an image using its K-L transform (图像的K-L变换展开), 110
expansion of an image in terms of vector outer products (图像的向量外积展开), 21 ~ 22
expected value of a random variable (随机变量的期望值), 92

F

false contouring (假轮廓), 2

fast Fourier transform (快速傅里叶变换), 84
feature map (特征图), 315 ~ 316
features (特征), 288
FFT (see fast Fourier transform)
filter (FFT (参看快速傅里叶变换) 滤波器), 155
Fourier transform (傅里叶变换), 45, 63, 72
frequency convolution theorem (频域卷积定理), 75
frequency sampling (频域采样), 172, 182 ~ 191
Fresnel integrals (Fresnel积分), 202 ~ 204

G

Gaussian noise (高斯噪声), 144
geometric progression (等比级数), 64
geometric restoration (几何复原), 193 ~ 198
grey level (灰度级), 1
grey level interpolation (灰度级插值), 195

H

Haar functions (哈尔函数), 46
Haar transforms (哈尔变换), 47, 62
Haar wavelet (哈尔小波), 62
Hadamard matrices (哈达玛矩阵), 57
Hadamard transforms (哈达玛变换), 57
highpass filter (高通滤波器), 148, 160
histogram equalization (直方图均衡化), 127
histogram hyperbolization (直方图夸张化), 129
histogram modification with random additions (随机增量式直方图修正法), 127, 130
histogram of an image (图像的直方图), 125
histogram of an image under variable illumination (可变光照下的图像直方图), 283 ~ 285
homogeneous random field (齐次随机场), 96
homomorphic filter (同态滤波器), 149 ~ 150
Hotelling transform (霍特林变换), 89
hysteresis thresholding (滞后阈值法), 266

I

ideal highpass filter (理想高通滤波器), 148, 160
ideal lowpass filter (理想低通滤波器), 148, 157 ~ 161
image(图像), 1
image as a linear superposition of point sources (作为点源的线性叠加的图像), 9
image as a random field (作为随机场的图像), 89 ~ 121
image classification (图像分类), 266
image compression (图像压缩), 18
image compression using K-L (基于K-L的图像压缩), 102
image compression using SVD (基于SVD的图像压缩), 24

image enhancement (图像增强), 18, 125 ~ 153
 image geometric restoration (图像几何复原), 193 ~ 198
 image labelling (图像标签法), 266
 image registration (图像对准), 193
 image resolution (图像分辨率), 2
 image restoration (图像复原), 18, 193 ~ 263
 image restoration by inverse filtering (用逆滤波使图像复原), 209 ~ 217
 image restoration by matrix inversion (用矩阵求逆使图像复原), 230 ~ 262
 image restoration by Wiener filtering (用Wiener滤波器使图像复原), 218 ~ 230
 image segmentation (图像分割), 265
 image sharpening (图像锐化), 148
 image smoothing (图像平滑), 147
 image thresholding (图像阈值), 255 ~ 286
 image thresholding under variable illumination (可变光照下的图像阈值法), 285
 impulse noise (脉冲噪声), 144
 independent random variables (独立随机变量), 92
 inhomogeneous contrast (非齐次对比度), 131

J

joint distribution function of many random variables (多个随机变量的联合分布函数), 92
 joint probability density function of many random variables (多个随机变量的联合概率密度函数), 92

K

Karhunen-Loeve expansion of an image (图像的K-L扩展), 110
 Karhunen-Loeve transform (K-L变换), 89
 Karhunen-Loeve transform of a multispectral image (多光谱图像的K-L变换), 136
 K-L (see Karhunen-Loeve)
 Kronecker ordering of Walsh functions (沃尔什函数的K-L (参看Karhunen-Loeve) Kronecker序), 57
 Kronecker product of matrices (矩阵的Kronecker乘积), 14, 237

L

La Vallee Poussin theorem (La Vallee Poussin定理), 180, 181
 Lagrange multipliers (拉格朗日乘子), 251 ~ 258
 Laplacian (拉普拉斯算子), 243
 least square error solution for image restoration (图像复原的最小二乘解), 218
 least square error approximation of an image (图像的最

小二乘逼近), 37x

Leibnitz rule for differentiating an integral with respect to a parameter (对参数积分求微分的莱布尼茨法则), 268

lexicographic ordering of Walsh functions (沃尔什函数的字典序), 57

limiting set of equations (等式的极限集), 180

line detection (线条检测), 309 ~ 310

linear operator (线性算子), 6

linear programming (线性规则), 172, 174

local contrast enhancement (局部对比度增强), 131

low pass filter (低通滤波器), 148, 157, 161

lowpass filtering (低通滤波), 147

M

matrix diagonalization (矩阵对角化), 24 ~ 35

maximizing algorithms for filter design (滤波器设计的极大化算法), 180

mean square error for K-L transform (K-L变换的均方差), 118

mean value of a random variable (随机变量的中值), 92
 median filter (中值滤波器), 146

minimizing algorithms for filter design (滤波器设计的极小化算法), 180

mini-max algorithms for filter design (滤波器设计的极小—极大化算法), 180

minimum error threshold (最小误差阈值), 268 ~ 278

minimum mean square error approximation of an image (图像的最小均方误差逼近), 118, 124

minimum square error approximation of an image (图像的最小平方误差逼近), 36

motion blurring (运动模糊), 200 ~ 204, 210 ~ 217, 228 ~ 230, 259 ~ 262

multiband image (多频带图像), 135

multiplicative noise (乘性噪声), 144, 149, 283

multispectral image (多光谱图像), 135

N

natural order of Walsh functions (沃尔什函数的自然序数), 47

nearest neighbour interpolation (近邻插值), 195

noise (噪声), 144

noise convolved with a filter (噪声与滤波器的卷积), 316 ~ 317

noise in image restoration (图像复原中的噪声), 210

non-maxima suppression (非极大抑制), 303

non-recursive filter (非递归滤波器), 161

norm of a matrix (矩阵范数), 35

O

- operator (算子), 6
- optimal threshold (最佳阈值), 268 ~ 278
- orthogonal matrix (正交矩阵), 24
- orthogonal random variables (正交随机变量), 92
- orthogonal set of functions (函数的正交集), 45
- orthonormal set of functions (函数的标准正交集), 45
- orthonormal vectors (标准正交向量), 24
- Otsu's thresholding method (otsu阈值方法), 278 ~ 282
- outer product of vectors (向量的外积), 21

P

- partition of a matrix (矩阵分块), 10
- pattern recognition (模式识别), 288
- pel (像素), 1
- pincushion distortion (针垫形失真), 194
- pixel (像素), 1
- point source (点源), 7 ~ 9
- point spread function of a linear degradation process (线性退化处理的点扩散函数), 198 ~ 209
- point spread function of an operator (算子的点扩散函数), 6 ~ 7
- principal component analysis of a multispectral image (多光谱图像的主分量分析), 136 ~ 144
- probability of an event (事件的概率), 90
- probability density function of a random variable (随机变量的概率密度函数), 91, 92
- probability density function of many random variables (多个随机变量的概率密度函数), 92
- properties of a discrete Fourier transform (离散傅里叶变换的性质), 79 ~ 84

Q

- quadtree (四叉树), 289

R

- ramp edges (斜边缘), 309
- random field (随机场), 90, 93
- random variable (随机变量), 90
- rank order filtering (排序滤波), 146
- rectangle function (矩形函数), 7
- recursive filter (递归滤波器), 161
- region growing (区域增长法), 288
- resolution (分辨率), 2
- restoration by matrix inversion (用矩阵求逆复原), 230 ~ 262
- restoration by Wiener filtering (用Wiener滤波器复原), 218 ~ 230

- restoration of motion blurring (用运动模糊复原), 210 ~ 217, 228 ~ 230, 259 ~ 262
- Rice's formula for filtered noise (滤波器去噪的Rice公式), 321
- Robinson operators (Robinson算子), 208

S

- salt and pepper noise (椒盐噪声), 144
- scaling function (伸缩函数), 62
- seed pixels for region growing (区域增长法的种子像素), 288
- separability assumption (可分离假设), 14 ~ 15
- separable masks (可分离模板), 306
- separable point spread function (可分离点扩散函数), 7
- separable transform (可分离变换), 15
- sequency order of Walsh functions (沃尔什函数的列率序), 47
- sharpening (锐化), 148
- shift invariant point spread function (位移不变的点扩散函数), 7
- shifting property of the delta function (delta 函数位移性质), 9
- simulated annealing (模拟退火), 217
- singular value decomposition of an image (图像的奇异值分解), 34
- singular value decomposition of a matrix (矩阵的奇异值分解), 24
- smoothing (平滑), 147
- Sobel masks (sobel 模板), 296 ~ 299, 306
- spatial statistics of a random field (随机场的空间统计), 97
- spatial autocorrelation matrix of an image (图像的空间自相关矩阵), 105
- spectral bands (频带), 1, 138
- spectral density of a random field (随机场的频谱密度), 223
- split and merge algorithms (拆分和合成算法), 288
- stability condition in filter design (滤波器设计的稳定条件), 171
- stacking operator (堆积算子), 10
- standard deviation of a random variable (随机变量的标准偏差), 92
- successive doubling algorithm (逐次加倍算法), 84
- SVD (see singular value decomposition) system function (SVD (参看奇异值分解) 系统函数), 155

T

- textured regions (纹理区域), 288

thresholding (阈值), 255
tie points (关节点), 195
time convolution theorem (时域卷积定理), 75
trace of a matrix (矩阵的迹), 36
transfer function from an astronomical image (天文图像的传递函数), 204
transfer function from a bright edge (明亮边缘的传递函数), 206 ~ 209
transfer function from a bright line (明亮线的传递函数), 205 ~ 206
transfer function of a degradation process (退化处理传递函数), 199 ~ 209
transfer function of motion blurring (运动模糊传递函数), 200 ~ 204, 210 ~ 217

U

uncorrelated data (不相关数据), 144
uncorrelated random fields (不相关随机场), 95
uncorrelated random variables (不相关随机变量), 92
unit sample response of a filter (滤波器的单位采样响应), 155
unitary matrix (酉矩阵), 23 ~ 24

unitary transform (酉变换), 23

V

variable illumination (可变光照), 149, 283 ~ 286
variance of a random variable (随机变量方差), 92
vector outer product (向量外积), 21
vector outer product (expansion of an image in terms of) (向量外积 (图像术语的扩充)), 21 ~ 22

W

Walsh functions (沃尔什函数), 46, 47
Walsh transforms (沃尔什变换), 47, 62
wavelets (小波), 62
white noise (白噪声), 220
Wiener filter (Wiener滤波器), 218 ~ 230, 254
Wiener-Khinchine theorem (Wiener-Khinchine 定理), 223
windowing (窗口法), 172, 173, 210

Z

z-transform (Z变换) 161 ~ 171